Using Liferay DXP

A Complete Guide

THE LIFERAY DOCUMENTATION TEAM
Richard Sezov, Jr.
Jim Hinkey
Stephen Kostas
Jesse Rao
Cody Hoag
Nicholas Gaskill
Michael Williams

Liferay Press

Using Liferay DXP 7.0
by The Liferay Documentation Team

Contents

Contents i

Preface ix
Conventions . . . ix
Publisher Notes . . . ix

I Using Liferay DXP

1 What is Liferay? 1
1.1 Building a site with Liferay Web Experience . . . 1
1.2 Using Liferay DXP as a collaborative platform . . . 6
1.3 Using Liferay as a web platform . . . 7
1.4 Extending and customizing Liferay for your own needs . . . 9
1.5 Summary . . . 10

2 Building Websites with Liferay Web Experience Management 11

3 Creating Basic Web Content 13
3.1 Initial Setup . . . 13
3.2 Using Liferay DXP's Content Editor . . . 15
3.3 Publishing Basic Content . . . 17

4 Creating a Site 21
4.1 Creating Pages with Layouts . . . 21
4.2 Building the Lunar Guides and Book a Trip Pages . . . 24
4.3 Adding Applications to Pages . . . 25
4.4 Page Templates . . . 26
4.5 Nesting Pages . . . 28
4.6 Site Navigation . . . 30

5 Creating Content 33
5.1 Creating More Content . . . 33
5.2 Using Documents and Media . . . 36

6 Web Content Structures and Templates **41**
6.1 Creating Structures . . . 41
6.2 Creating Templates . . . 45
6.3 Creating Content with Structures and Templates . . . 48
6.4 Advanced Templates . . . 49
6.5 Publishing Content with Content Display Pages . . . 54
6.6 Advanced Content Display Page Options . . . 56
6.7 Application Display Templates . . . 58
6.8 Review Your Site . . . 63

7 Web Experience Management **65**

8 Starting Site Development **67**
8.1 Creating Sites . . . 67
8.2 Creating and Managing Pages . . . 78
8.3 Displaying Pages on Mobile Devices . . . 93
8.4 Building Sites from Templates . . . 97
8.5 Creating Pages from Templates . . . 102
8.6 Importing/Exporting Pages and Content . . . 105
8.7 Creating Teams for Advanced Site Membership Management . . . 108

9 Creating Web Content **111**
9.1 Publishing Basic Web Content . . . 112
9.2 Designing Uniform Content . . . 126

10 Staging Content for Publication **141**

11 Enabling Staging **143**
11.1 Enabling Local Live Staging . . . 143
11.2 Enabling Remote Live Staging . . . 144
11.3 Enabling Page Versioning and Staged Content . . . 148
11.4 Publishing Staged Content Efficiently . . . 150
11.5 Using the Staging Environment . . . 152
11.6 Using Site Pages Variations . . . 157
11.7 Scheduling Web Content Publication . . . 159
11.8 What's Expected from Staging? . . . 161

12 Publishing Content Dynamically **199**
12.1 Organizing Content with Tags and Categories . . . 199
12.2 Defining Content Relationships . . . 203
12.3 Geolocating Assets . . . 206
12.4 Publishing Assets . . . 211
12.5 Publishing RSS Feeds . . . 219
12.6 Restoring Deleted Assets . . . 225

13 Configuring Applications **231**
13.1 Look and Feel Configuration . . . 231
13.2 Exporting/Importing App Data . . . 234
13.3 Communication Between Apps . . . 238

13.4 Sharing Applications with Other Sites . . . 240
13.5 Application Permissions . . . 242
13.6 Application Scope . . . 242
13.7 Configuration Templates . . . 243
13.8 Styling Apps with Application Display Templates . . . 245

14 Collaboration **251**

15 Managing Documents and Media **253**

16 Publishing Files **255**
16.1 Adding Files to a Document Library . . . 255
16.2 Editing Images . . . 264
16.3 Accessing Google Docs™ . . . 266
16.4 Displaying Files . . . 269
16.5 Collaborating on Files . . . 271
16.6 Publishing Files Summary . . . 276
16.7 Using Document Types and Metadata Sets . . . 276
16.8 Liferay Store Types . . . 280

17 Using a SharePoint Repository **285**
17.1 Creating a SharePoint Application . . . 286
17.2 Creating a New SharePoint Repository Configuration . . . 287
17.3 Adding a SharePoint Documents and Media Repository . . . 289
17.4 Using External Repositories . . . 289
17.5 Administering Liferay Sync . . . 296
17.6 Using Liferay Sync on Your Desktop . . . 302
17.7 Using Liferay Sync on Your Mobile Device . . . 312

18 Adapting Your Media Across Multiple Devices **319**
18.1 Installing Adaptive Media . . . 319
18.2 Adding Image Resolutions . . . 322
18.3 Managing Image Resolutions . . . 323
18.4 Creating Content with Adapted Images . . . 325
18.5 Migrating Documents and Media Thumbnails to Adaptive Media . . . 327
18.6 Advanced Configuration Options . . . 328

19 Collaborating **331**
19.1 Publishing Blogs . . . 331
19.2 Creating Forums with Message Boards . . . 341
19.3 Mentioning Users . . . 355
19.4 Working Together with the Wiki . . . 357
19.5 Sending Alerts and Announcements . . . 371
19.6 Managing Notifications and Requests . . . 373
19.7 Informing Users with the Knowledge Base . . . 375
19.8 Bookmarking Sites . . . 396
19.9 Summary . . . 399
19.10 Sharing Status with Microblogs . . . 399
19.11 Inviting Members to Your Site . . . 402

19.12 Sending Messages with Private Messaging . . . 404
19.13 Managing Subscriptions . . . 405

20 Creating A Social Network **407**
20.1 Leveraging Social Applications, Activities Tracking, and User Connections . . . 407
20.2 Measuring Social Activity . . . 412
20.3 Exporting Liferay DXP Applications To Other Websites . . . 413
20.4 Integrating with Facebook, Twitter, and More . . . 415

21 Search **419**
21.1 Searching the Index, not the Database . . . 419
21.2 Leveraging Elasticsearch in Liferay DXP . . . 420
21.3 Liferay DXP Search Features . . . 420
21.4 Searching for Assets . . . 422
21.5 Search Results . . . 425
21.6 Configuring Facets . . . 431

22 Forms and Lists **437**
22.1 Which Form Builder Should I Use? . . . 437

23 Collecting Information from Users **439**
23.1 Basic Forms . . . 439
23.2 Creating Advanced Forms . . . 449
23.3 Sending Form Entries through a Workflow . . . 464

24 Creating Simple Applications **471**
24.1 Creating Data Definitions . . . 471
24.2 Creating Data Lists . . . 485
24.3 Using Templates to Display Forms and Lists . . . 488

25 Using Workflow **499**
25.1 Embedded Workflows . . . 499
25.2 Workflow Assignees Have Implicit Asset Permissions . . . 500
25.3 Enabling Workflow . . . 500

26 Kaleo Designer **505**
26.1 Managing Workflows with Kaleo Designer . . . 505
26.2 Workflow Definition Nodes . . . 508
26.3 Affecting the Processing of Workflow Definitions . . . 513
26.4 Creating Tasks in Kaleo Designer . . . 516
26.5 Workflow Forms . . . 520

27 User Management **531**
27.1 Adding and Managing Users . . . 531
27.2 Adding and Managing Organizations . . . 539
27.3 User Groups . . . 546
27.4 Roles and Permissions . . . 557

28 GDPR Tools **565**

28.1 Anonymizing Data . . . 565
28.2 Manual Anonymization . . . 567
28.3 Sanitizing User Data . . . 567
28.4 Exporting User Data . . . 572
28.5 Auditing Users . . . 578
28.6 Password Policies . . . 585

29 Managing Apps **589**
29.1 Managing and Configuring Apps . . . 589
29.2 Using the Liferay Marketplace . . . 592
29.3 Installing Apps Manually . . . 597
29.4 Liferay App Types . . . 599
29.5 Blacklisting OSGi Modules and Components . . . 600

30 Configuring Liferay DXP **605**
30.1 Where Configuration Happens . . . 605
30.2 Configuration Scope . . . 605

31 System Wide Settings **607**
31.1 System Settings . . . 607
31.2 Understanding System Configuration Files . . . 611
31.3 Environment Variables . . . 615
31.4 Server Administration . . . 618

32 Setting Up a Liferay DXP Instance **625**
32.1 Virtual Instances . . . 625
32.2 Instance Settings . . . 627

33 Using Liferay's Script Engine **633**
33.1 Invoking Liferay Services From Scripts . . . 633
33.2 Running Scripts From the Script Console . . . 634
33.3 Leveraging the Script Engine in Workflow . . . 641
33.4 Using Web Services for Remote Portlets (WSRP) . . . 644
33.5 Configuring Form Navigator Forms . . . 647
33.6 Custom Fields . . . 650

34 Targeting Content To Your Audience **655**
34.1 Managing User Segments . . . 656
34.2 Managing Campaigns . . . 661
34.3 Using the Audience Targeting Applications . . . 668
34.4 Simulating User Segments and Campaigns . . . 672
34.5 Liferay Audience Targeting Rules . . . 673
34.6 Audience Targeting Metrics . . . 676
34.7 Audience Targeting System Settings . . . 677

II Deploying Liferay DXP **679**

35 Introduction to Liferay DXP Deployment **681**

35.1 Supported Platforms and Limitations . . . 681
35.2 Reference Architecture . . . 681

36 Deploying Liferay DXP **685**
36.1 Preparing for Install . . . 685
36.2 Installing Liferay DXP . . . 690
36.3 Installing Liferay DXP Manually . . . 694
36.4 Installing Liferay DXP on Tomcat 8 . . . 697
36.5 Installing Liferay DXP on Wildfly . . . 704
36.6 Installing Liferay DXP on JBoss EAP 6.4 . . . 711
36.7 Installing Liferay DXP on tc Server . . . 719
36.8 Installing Liferay DXP on WebLogic 12c R2 . . . 725
36.9 Installing Liferay DXP on WebSphere 8.5.5 . . . 729
36.10 Installing Elasticsearch . . . 739
36.11 Liferay DXP Trial Plugin Installation . . . 743
36.12 Activating Liferay DXP . . . 746
36.13 Setting Up Marketplace and Portal Security . . . 747
36.14 Choosing IPv4 or IPv6 . . . 748

37 Configuring Liferay DXP **749**
37.1 Locales and Encoding Configuration . . . 749
37.2 Document Repository Configuration . . . 751
37.3 Liferay DXP Clustering . . . 757

38 Updating a Cluster **767**
38.1 Using Rolling Restarts . . . 767
38.2 Other Cluster Update Techniques . . . 769
38.3 Configuring Remote Staging in a Clustered Environment . . . 770

39 Liferay Digital Enterprise Configuration and Tuning Guidelines **775**
39.1 Application Server Tuning . . . 775
39.2 Java Virtual Machine Tuning . . . 777
39.3 Content Delivery Network . . . 778

40 Liferay DXP Security **783**
40.1 Liferay DXP Security Overview . . . 783

41 What is SAML? **787**
41.1 Important SAML URLs . . . 787
41.2 Single Sign On . . . 788
41.3 Single Log Off . . . 791
41.4 Related Topics . . . 793
41.5 Setting up Liferay DXP as a SAML Identity Provider . . . 794
41.6 Setting up Liferay DXP as a SAML Service Provider . . . 799
41.7 Configuring SAML . . . 803
41.8 Logging in to Liferay DXP . . . 804
41.9 Service Access Policies . . . 805
41.10 Authentication Verifiers . . . 809
41.11 LDAP . . . 810

41.12 Token-based Single Sign On Authentication . . . 816
41.13 Authenticating with OpenID Connect . . . 818
41.14 OpenID Single Sign On Authentication . . . 820
41.15 CAS (Central Authentication Service) Single Sign On Authentication . . . 822
41.16 OpenAM Single Sign On Authentication . . . 823
41.17 Facebook Connect Single Sign On Authentication . . . 825
41.18 NTLM Single Sign On Authentication . . . 827
41.19 OAuth . . . 829
41.20 AntiSamy . . . 831
41.21 Authenticating with Kerberos . . . 832

42 Enabling Antivirus Scanning for Uploaded Files 837

43 Maintaining Liferay DXP 841

44 Patching Liferay DXP 843

45 Patching Basics 845
45.1 Fix Packs . . . 845
45.2 Hotfixes . . . 845
45.3 Service Packs . . . 845
45.4 How Patches are Tested . . . 846

46 Configuring the Patching Tool 847
46.1 Patching Tool Basic configuration . . . 847
46.2 Patching Tool Advanced Configuration . . . 848
46.3 Installing patches on the 7.0 WAR . . . 849

47 Using the Patching Tool 853
47.1 Executables . . . 853
47.2 Installing Patches . . . 853
47.3 Removing or Reverting Patches . . . 855
47.4 Cleaning Up . . . 855
47.5 Comparing Patch Levels . . . 856
47.6 Showing collisions between patches and deployed plugins . . . 856
47.7 Separating the Patches from the Liferay DXP Installation . . . 856

48 Keeping up with Fix packs and Service Packs 859
48.1 Securing Liferay DXP's Remote Services . . . 859

49 Backing up a Liferay DXP Installation 863
49.1 Backing up Source Code . . . 863
49.2 Backing up Liferay's File System . . . 863
49.3 Backing up Liferay DXP's Database . . . 864

50 Upgrading to 7.0 865
50.1 Pre upgrade - Speed up the process . . . 865
50.2 Preparing an Upgrade to 7.0 . . . 870
50.3 Running the Upgrade Process . . . 873

50.4 Upgrading a Sharded environment . . . 881
50.5 Upgrading Social Office . . . 883

51 Monitoring Liferay DXP **889**
51.1 Monitoring Garbage Collection and JVM . . . 889
51.2 Advanced Monitoring: APM Tools - Dynatrace . . . 891

52 Liferay Enterprise Search **901**
52.1 Preparing to Install Elasticsearch . . . 901
52.2 Configuring Elasticsearch . . . 902
52.3 Securing Elasticsearch with Shield . . . 913
52.4 Tuning and Scaling Elasticsearch . . . 918
52.5 Installing Marvel for Elasticsearch . . . 918
52.6 Using Solr . . . 922

53 Backing Up Liferay Enterprise Search **931**
53.1 Backing Up Elasticsearch . . . 931
53.2 Upgrading to Elasticsearch 6 . . . 933
53.3 Securing Elasticsearch 6.5 with X-Pack . . . 936
53.4 Installing X-Pack Monitoring for Elasticsearch 6.5 . . . 939

54 Securing Elasticsearch 6.1 with X-Pack **943**
54.1 Installing X-Pack . . . 943
54.2 Setting Up X-Pack Users . . . 944
54.3 Enabling Transport Layer Security . . . 945
54.4 Install and Configure the Liferay Enterprise Search Security app . . . 946
54.5 Installing X-Pack Monitoring for Elasticsearch 6.1 . . . 947
54.6 Monitoring in Liferay DXP . . . 950

55 Managing Liferay DXP with Liferay Connected Services **951**
55.1 LCS Preconfiguration . . . 952
55.2 Registering Your Liferay DXP Server with LCS . . . 959
55.3 Using LCS . . . 963
55.4 Troubleshooting Your LCS Connection . . . 982
55.5 Building Clustering for Liferay Portal . . . 988
55.6 Configuration Reference . . . 988
55.7 Database Templates . . . 988
55.8 Elasticsearch Settings . . . 989
55.9 Form Field Types . . . 990
55.10 Deployment Reference . . . 993
55.11 Knowledge Base Configuration . . . 994
55.12 Shield Adapter Settings . . . 1002
55.13 Solr Settings . . . 1003

Preface

Welcome to the world of Liferay DXP! This book was written for anyone who has any part in setting up, using, or maintaining a web site built on Liferay DXP. For the end user, it contains everything you need to know about using the applications included with Liferay. For the administrator, you'll learn all you need to know about setting up your site with users, sites, organizations, and user groups, as well as how to manage your site's security with roles. For server admins, it guides you step-by-step through the installation, configuration, and optimization of Liferay DXP, including setting it up in a clustered, enterprise-ready environment. Use this book as a handbook for everything you need to do to get your Liferay DXP installation running smoothly, and then keep it by your side as you configure and maintain your Liferay-powered web site.

Conventions

The information contained herein has been organized in a way that makes it easy to locate information. The book has two parts. The first part, *Using Liferay Portal*, describes how to configure and use a freshly installed Liferay Portal. The second part, *Deploying Liferay Portal*, is for administrators who want to install Liferay Portal and optimize its performance.

Sections are broken up into multiple levels of headings, and these are designed to make it easy to find information.

Source code and configuration file directives are presented monospaced, as below.

```
Source code appears in a non-proportional font.
```

Italics represent links or buttons to be clicked on in a user interface.
`Monospaced type` denotes Java classes, code, or properties within the text.
Bold describes field labels and portlets.
Page headers denote the chapters and the section within the chapter.

Publisher Notes

It is our hope that this book is valuable to you, and that it becomes an indispensable resource as you work with Liferay DXP. If you need assistance beyond what is covered in this book, Liferay offers training[1], consulting[2],

[1] https://learn.liferay.com
[2] https://www.liferay.com/consulting

and support[3] services to fill any need that you might have.

For up-to-date documentation on the latest versions of Liferay, please see the documentation pages on Liferay Learn.[4]

As always, we welcome feedback. If there is any way you think we could make this book better, please feel free to mention it on our forums or in the feedback on Liferay Learn. You can also use any of the email addresses on our Contact Us page.[5] We are here to serve you, our users and customers, and to help make your experience using Liferay DXP the best it can be.

[3] https://help.liferay.com
[4] https://learn.liferay.com
[5] https://www.liferay.com/contact-us

Part I

Using Liferay DXP

Chapter 1

What is Liferay?

Let's attempt some mind reading: you're reading this because you need to build a website. You're thinking about using Liferay Digital Experience Platform for your site, or you've already decided to use it and you want to learn all about it. Either way, you've come to the right place. What you'll find here is that Liferay DXP is the right decision, and we'll be happy to tell you all the reasons why. But since you might be in a rush, we'll give you all the reasons in a nutshell right here, and then you can read the details in the rest of the articles.

The reasons to use Liferay DXP for your website are simple: it provides a robust *platform* to build your site on quickly and serve it to all clients, be they desktop, mobile, or anything in between; it provides all the standard *applications* you need to run on your site; and it provides an easy to use development *framework* for new applications or customization. In addition to this, Liferay DXP is developed using an open source methodology, by people from around the world. The code base is solid and has been proved to be reliable and stable in mission critical deployments in diverse industries.

But don't just take our word for it. Let us *show* you how Liferay does all this. Probably the first thing you want to do after you install Liferay DXP to manage your site is to get your content published. Let's dive right in and see how Liferay DXP handles that task.

1.1 Building a site with Liferay Web Experience

When you log into Liferay DXP and look at its default screen, one thing to notice is that it's built for all clients that access the web, not just desktop browsers.

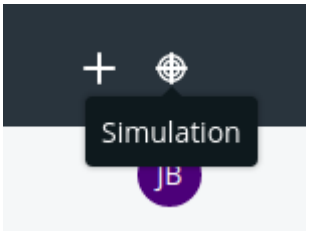


Figure 1.1: 1: If you click this button, a preview appears that lets you see how the page would look when displayed at various device resolutions.

Another thing to notice is that all the page controls are right there for you. For example, to add content or applications to the page, click the + button, and then you can add any kind of content Liferay DXP supports. The interface has been designed to get out of your way and let you do your work.

Liferay Web Experience Management scales to work for the tiniest of sites all the way up to the largest of sites. For example, you can click that *Add* button, choose *Add New → Basic Web Content*, and immediately start typing content into a WYSIWYG editor, in place. Or you can set up Liferay to host many different web sites, all with their own domain names. Each site can take advantage of a separate staging server, where content and pages are created by teams of people using structures and templates, and updates to the production server are published on a schedule, only after having gone through a multi-step approval process.

That's powerful.

By default, Liferay DXP starts with a single site that has a single page. You can build any website you wish out of this, complete with multi-nested page hierarchies, as the figure below shows.

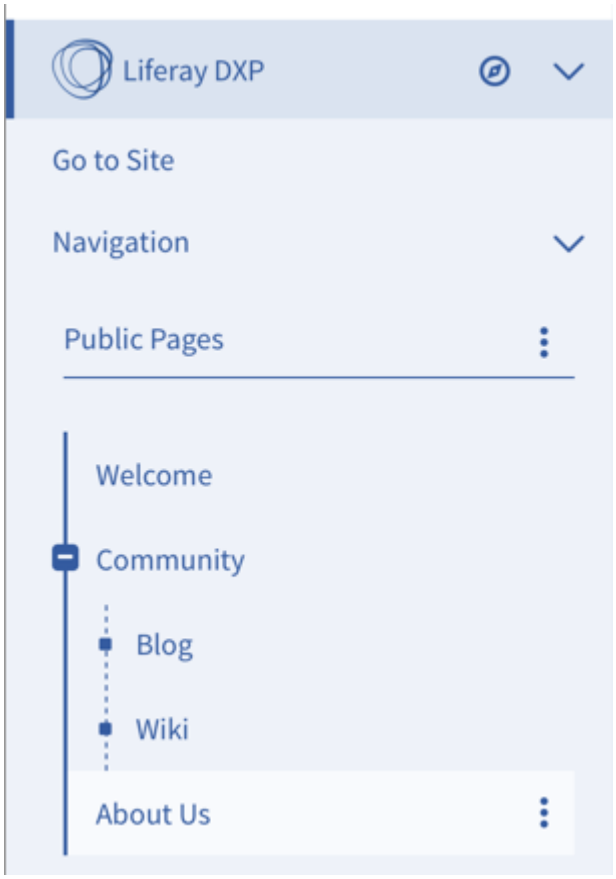


Figure 1.2: 2: Liferay's page hierarchies are easy to create, using a tree structure that's familiar to anyone who has used a file manager.

These pages can have any layout you like: Liferay DXP ships with several built-in, and you can create your own custom layouts and deploy them easily. Pages can be added, removed, or reordered any time, and you have the full flexibility of all the HTML page attributes, such as meta tags and robot file declarations, that you need.

Pages are also integrated with Liferay's powerful permissions system, so it's easy to restrict access to certain portions of your site. You can give individual users sites of their own, with public pages that have their content and blog, and private pages that contain their calendars and email.

If you're running a large website that has lots of different sub-sites for individuals and groups, you can use page templates and site templates. The former enables you to set up templates of pages with predefined layouts and applications already on them, and the latter enables you to create a whole site made up of multiple, predefined pages.

There's even more. If you have a very large site, you might need multiple people to work on it. And you certainly don't want the live site changing before your users' eyes. For that reason, Liferay DXP provides a feature called *staging* that lets you place your changes in a holding area while they're being worked on. You can have a local staging server, where the staged site resides on the same server as the live site, or you can have a remote staging server, where all web content work happens on a separate server from your live site. In either case, when you're ready, site changes can be pushed to the live site, either manually or on a schedule.

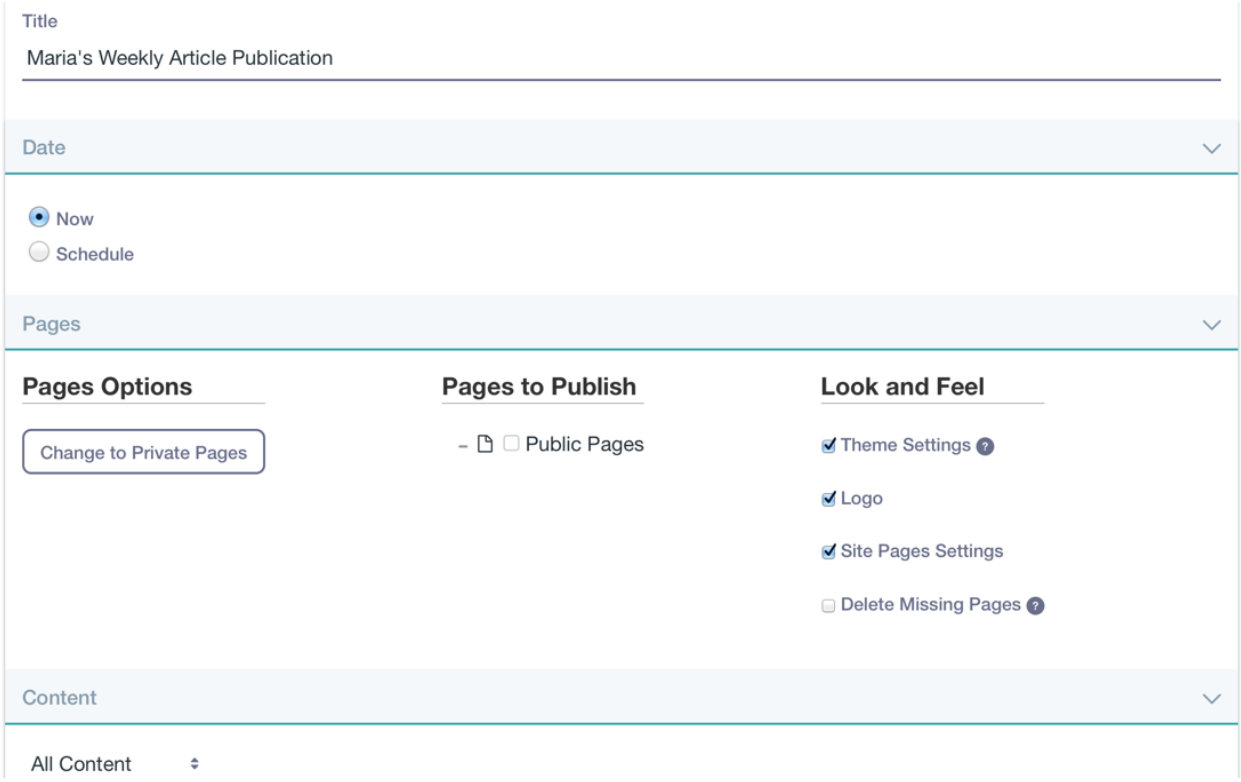


Figure 1.3: 3: Staging supports publishing manually or on a schedule.

Liferay DXP's web content creation tools are easy and intuitive to use at all levels. If you need only basic content management capabilities for your site, you can jump right in. You can add new content from the Add menu and drag it right into place. It's easy to go from this basic level of content management to more sophisticated levels of functionality.

For example, suppose you wanted to build a news-oriented site. Most of the content you'll publish is an article of some kind. Liferay's web content management system lets you create a *structure* for this, so that you can capture all the information from your writers that you'd need in an article. The figure below shows what this structure might look like to a journalist who'd be entering his or her article into the system.

As you can see, you can use structures to make sure writers provide the title of the story, what type of story it will be, and the byline (i.e., the writer's name). You've made sure that all the relevant information for the story is captured in the system.

Web content is one example of an *asset*. Assets are meta-data attached to content types, and that meta-data is used to aggregate similar assets together in searches or as published content. One way to do this in the example above is to tag and categorize stories so they can be found more easily by users.

This is just one example, of course. But the concept is applicable to any kind of site you'd want to build. For example, if you were building a site for a zoo, you could use web content structures to help users enter

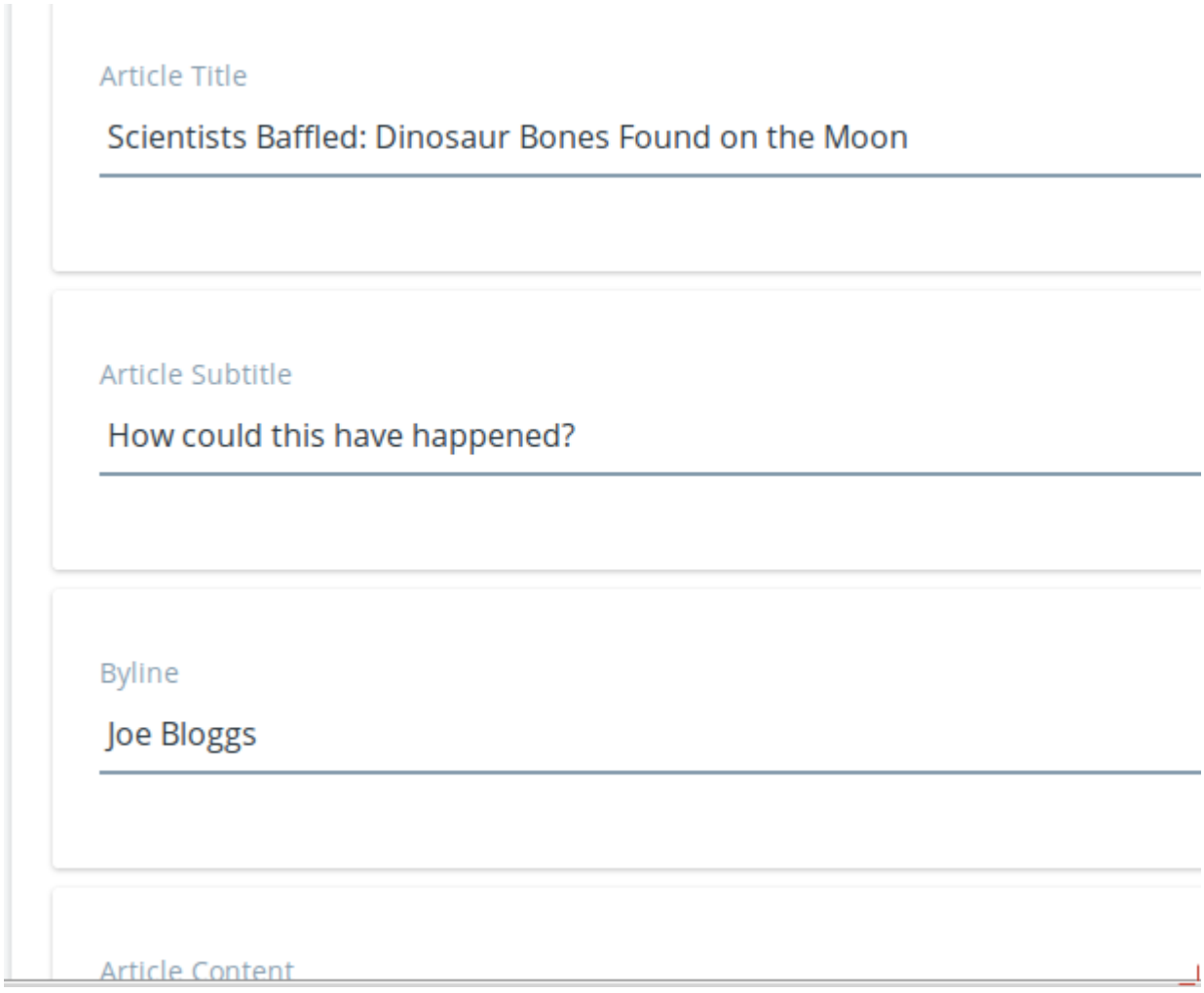


Figure 1.4: 4: Structures allow you to specify exactly the type of data that makes up your content. You can also include tooltips to help your users understand what each field is for.

data about animals in the zoo, such as their common names, their scientific names, their species, their locations in the wild, and more.

When it comes time to publish content, structures are combined with *templates*. Templates are instructions for how to display structures, written in Freemarker, a well-known templating language used for mixing HTML with programmatic elements. Because of this, templates are very easy to write and can help you ensure that your site has a consistent look and feel.

There is much more to web content. You can create abstracts, schedule when content is published and when it should be taken down (or reviewed), define related assets, and more.

This is just the web content portion of Liferay's content management system. Liferay DXP is also great at managing file-based content.

Keeping track of documents, images, video, and more

It's rare to find a full-featured content management system in an open source project. Most of the time, you'll find web content management systems and file-based content management systems as separate projects. Liferay DXP, however, provides you with both. As shown above, the web content management system is as

robust as any other you'll find, and its file-based content management system is the same.

You'll find Liferay's file-based content management system in an application called *Documents and Media Library*. This application resides on the Site Administration page or can be added to any page, and, as shown below, looks very much like the file manager that you're already familiar with from your operating system.

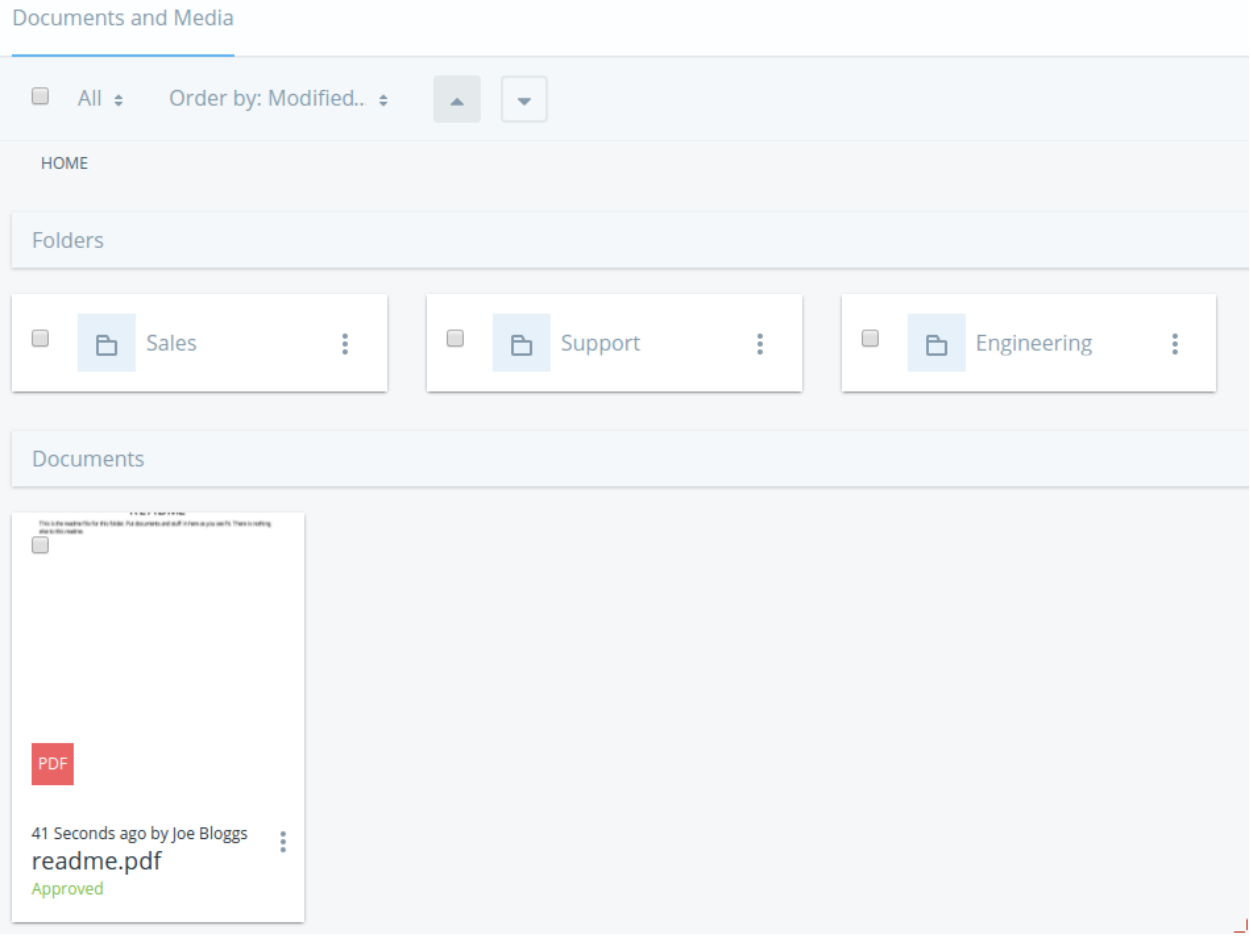


Figure 1.5: 5: Liferay DXP's Documents and Media library was purposefully designed to be familiar to anyone who uses a computer.

Like a file manager, you can browse files and folders in nested hierarchies. You can also mount other repositories that you might have in your environment, such as any system that implements Content Management Interoperability Services (CMIS) or Documentum (enterprise subscribers only). It provides previews of just about every document type you can think of. And, like a file manager, you can upload, copy, and move files between folders by dragging and dropping them. Of course, if you still want to use your operating system's file manager, you can, because Liferay's Documents and Media library supports WebDAV, using the same credentials you use to log in to Liferay.

Liferay DXP's Documents and Media library, however, is much more robust than a file manager is, because it's a full content management system. You can define ways of classifying files that may be of different types, but are meant for the same, overarching purpose.

For example, *metadata sets* are groups of fields describing attributes of a file. One that ships with the product is called *meeting metadata*, and it contains fields such as Meeting Name, Date, Time, Location, Description, and Participants. This is a generic set of fields that go together and that you'd want to use as a

group. You can create as many of these as you want.

For files, you can define *document types*. They provide a more natural way of working with files. For example, you might create a document type called Meeting Minutes. The file format doesn't matter: whether it's a Microsoft Word document, an HTML file, or a text file, the document contains meeting minutes. Once you've created the document type, you can attach the Meeting Metadata set that contains many of the fields you'd want, and you can also add extra fields, such as a field for action items. When users want to add files containing their notes for meeting minutes, they can also add all the relevant metadata about the meeting (such as the time, location, and action items). This captures the context information that goes with the document, and it provides a much more natural way of working with documents than just dumping them into a shared file system.

Of course, the system goes much further than this. Folders can be set so that only certain document types can be added to them. Workflow rules can also be added to folders to run files through an approval process that you define. In short, Liferay's file-based content management system gives you all the features you need to manage and share files in a group.

Many Liferay DXP users see it as a robust content management system, and they use it primarily for that purpose. Now, hopefully, you can see why. We'll cover the system in-depth in the relevant section on the left, but for now we need to look at some of the other ways you can use Liferay DXP, starting with its fantastic collaborative tools.

1.2 Using Liferay DXP as a collaborative platform

Many sites have grown organically. You may have grown your community by using separate tools: first a forums application, and then a wiki for collaborative documentation, and maybe even a chat application. It can be hard (and error-prone) to integrate all these applications so your users can use them seamlessly. Thankfully, Liferay includes a suite of collaborative applications, and they're all integrated together.

Liferay DXP offers every standard collaborative application that's available. These applications range from personal productivity applications like a calendar and email, to community-building applications like message boards, polls, and wikis.

This is a suite of integrated applications with all the features of similar, standalone applications. For example, Liferay DXP's message boards include categories and subcategories, message threads, captcha, RSS feeds, email notification, posting via email, and much more. But more than this, the applications are integrated with the rest of Liferay DXP's framework. Users log in and their profiles are used automatically by the message boards and all the other collaborative applications. And as we'll see later, functionality from the built in applications can be added to your own to provide features like comments in your own software, and you don't have to write any code to do it.

Liferay DXP's wiki is another example of a full-featured collaborative application. It has support for authoring pages in a WYSWYG editor, or more advanced users can use the easy-to-learn standard Wiki Creole syntax. The wiki keeps a full history of every change that's been made, allowing users to revert back to any change. Users can comment on wiki articles or read them via RSS feeds (just about every Liferay application supports this). Each site can have one or more wikis, and each wiki can have one or more top-level nodes.

One important feature of all the collaborative applications–as well as web content and documents–is the Recycle Bin. If users delete content that needs to be restored later, you don't have to find it in your backups: it's in the Recycle Bin.

Liferay DXP's suite of collaborative applications includes a blog (complete with blog aggregation features so you can publish multiple users' blog entries in one place), a chat application for users who are online at the same time, message boards, a wiki, a knowledge base that you can use to publish a library of articles, a

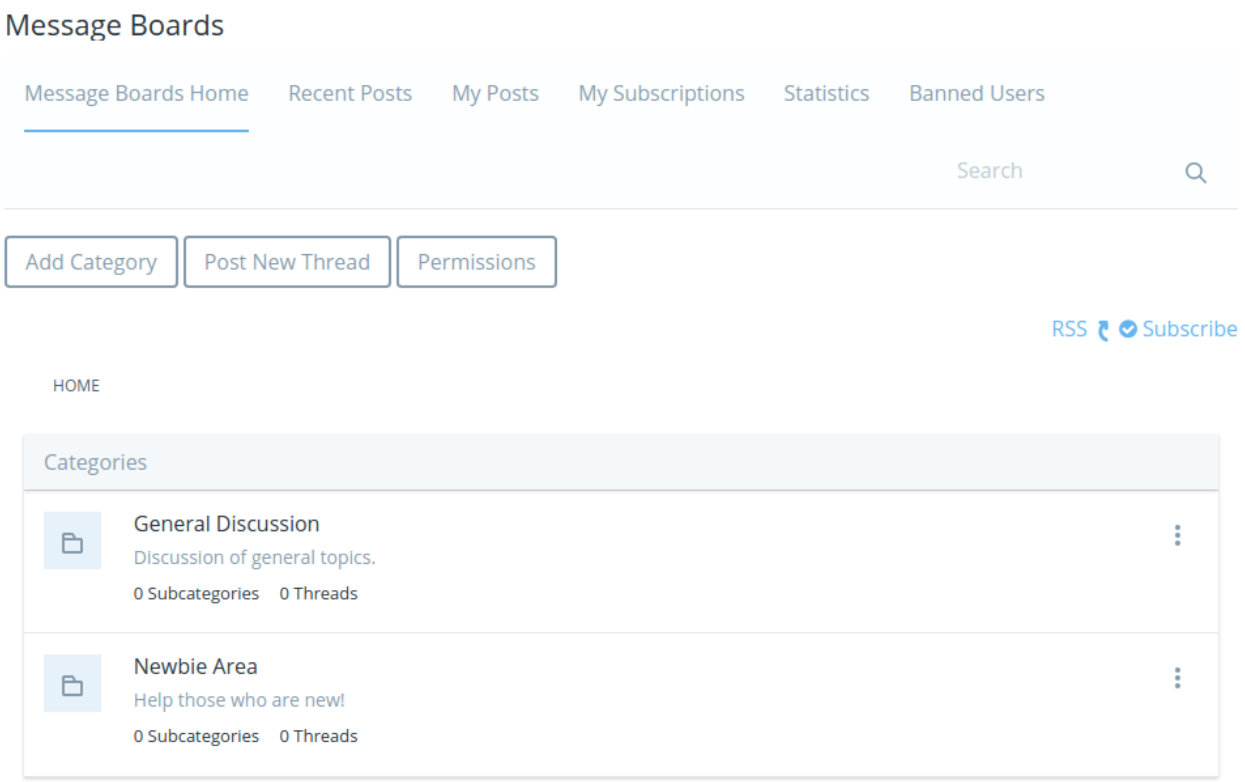


Figure 1.6: 6: Liferay DXP's message boards are as fully featured as any standalone forum application, with the added benefit that they're integrated with the rest of the system.

polling system you can use to have users vote on certain questions, and personal productivity applications like a calendar.

Liferay DXP includes every application you'll need to enable users to collaborate. Next, you'll see how Liferay can host multiple web sites.

1.3 Using Liferay as a web platform

We can't even begin to imagine what you're thinking of building, but whatever it is, you're going to put your heart and soul into it. Building it on Liferay's web platform can give you a leg up. It provides everything you need to support your application, so you can concentrate solely on what *you're* building, and not the rest of the features your users expect to come along with it.

Imagine your application for a moment. Does it require users to register on your site? Can users comment on content contained in your application? Is there something that users can tag or categorize? If you think about the layout of the application, would it benefit from modularization? Could you make use of a rich JavaScript framework with many components built into it? How about security–will you need to make information available to some users, but not to all users? Liferay DXP has all of this and more available to developers.

Liferay DXP's development framework is a great help when you're building an application for the web, for mobile, or for anything in between. For example, bug fixes to your applications are easy to apply, because Liferay applications are hot deployed to the running server. Liferay's Marketplace gives you a ready-made

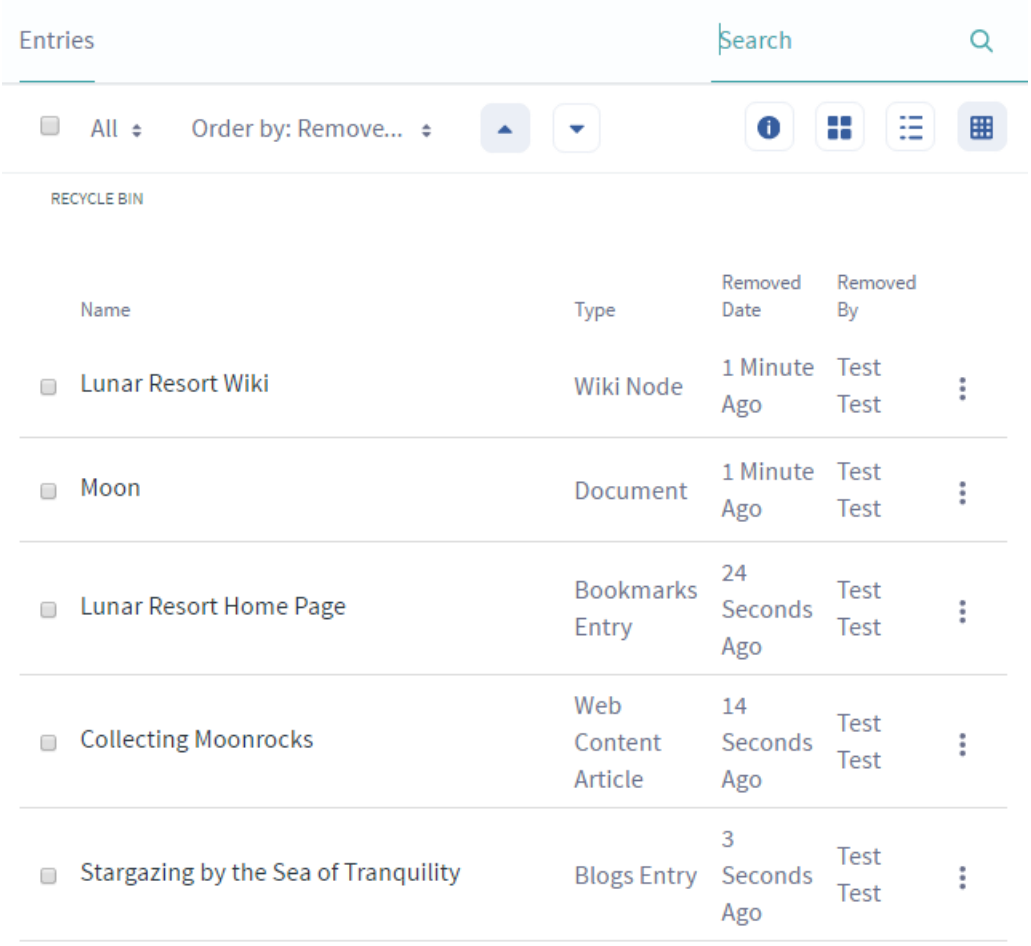


Figure 1.7: 7: The Recycle Bin can hold any kind of content.

shopping center for your applications. And Liferay's web services and JSON architecture make it easy for you to share data from your applications with other systems running on different platforms.

You get all this and much more. It's a very powerful platform, and certainly worth your investigation. Read the developer section of this site to learn more.

A great integration platform

If you're building an enterprise system, portals were designed in the first place to be a single point of entry to your users' applications and content. Since Liferay DXP integrates well with user directories such as LDAP and Active Directory, along with single sign-on systems such as SAML and OpenSSO, it fits well into your enterprise systems. This allows you to use it as an integration platform for existing applications.

Liferay DXP, since it adheres to the JSR standard for portlets, was designed from the ground up for application integration. You can add any application installed on the system to any page in the portal. You can make use of APIs provided by other systems to integrate their data into an application window in Liferay. And applications you create with Liferay's Service Builder framework are web service-enabled from the start.

Hosting multiple sites on Liferay DXP

Liferay DXP excels as a multi-site hosting platform. You can use it to host multiple sites under the same overall architecture, or you could host several completely different websites based solely on Liferay's ability to serve multiple instances of itself from the same physical installation.

In the first scenario, Liferay DXP's Sites architecture lets you create multiple, different websites that have public and/or private sets of pages and as many pages within those sets as you'd like. Users can join and leave open sites with one click. Alternatively, sites can be defined as restricted or private, and users can't access those unless they're added by site administrators. All these sites can have canonical domain names such as baseballcards.liferay.com or progrock.liferay.com.

Using this construct, you can build anything you can imagine. There is no limit to the number of sites you can have: some Liferay installations have only one or two, but others have many thousands. For those larger installations, Liferay DXP contains a complete site membership management framework that lets administrators manage automatic site members for groups of users. It really is built to scale to the size you need.

In the second scenario, Liferay DXP lets you create completely separate instances of itself from the same installation. Users, groups, organizations, sites, and roles from each instance are kept completely separate. If a user registers for a user id on one instance, he or she would have to register as a new user on one of the other instances.

This lets you host many different, separate websites from one Liferay DXP installation. Users of each instance have access to the same powerful content management, collaboration, social, and web development platform that they'd have if they were operating from a single, standalone installation.

Let's see how you can customize Liferay DXP so that it looks and operates exactly the way you've envisioned for your site.

1.4 Extending and customizing Liferay for your own needs

Beyond using it as a development platform for new applications, Liferay DXP has also been designed to be extended and modified. As an open source project, its source code is available, but Liferay DXP's developers have designed the product to make it easy to build whatever you want out of it.

The first (and easiest) way of customizing parts of Liferay DXP is with Application Display templates. These let you change the way built-in applications look. For example, if you don't like the Documents and Media Library's file manager view with large icons, you can create an Application Display template that shows documents in some other view. If you don't like the layout of the Blogs application, you can change it so that it has the look you want.

Liferay DXP goes far beyond this, though. Special software components called *modules* enable developers to change Liferay's interface and behavior–without having to modify any of Liferay DXP's source code. This provides you all the benefits of building your site from scratch, but without all the effort to actually build from scratch. If you want to make a change to the user registration screens, add support for a proprietary single sign-on mechanism that you've written, add a feature to the message boards application or anything else, you can make those customizations. And if you're a developer, you know that it's a whole lot easier to customize something that almost does things exactly the way you want than it is to write that feature from scratch. With Liferay DXP, you *can* have your cake and eat it too.

1.5 Summary

So what is Liferay? As you can see, it's hard to describe, because it does so much. What we've essentially done is say it's a totally awesome content and document managing, user collaborating, socially enabling, application developing, corporate integrating, completely customizable platform for building the Internet. If we'd said that up front, you'd probably have doubted us. Hopefully now, you can see that it's true.

If you're interested in using Liferay DXP for *your* product, continue reading. All of these features (and more that we couldn't mention) are covered here.

Chapter 2

Building Websites with Liferay Web Experience Management

Nowadays, few people build websites using only a text editor because that still requires a lot of effort. Why do it manually when you can have the computer do a lot of the work for you? With Liferay Web Experience Management, you can focus on your site and let Liferay DXP handle those low-level details. It provides web content management, document management, and a framework for robust content display and advanced content creation. With so many features, it's tough to know where to start. This is a step by step guide to creating content, pages, and sites, starting with the most basic elements and ending with a full-featured website.

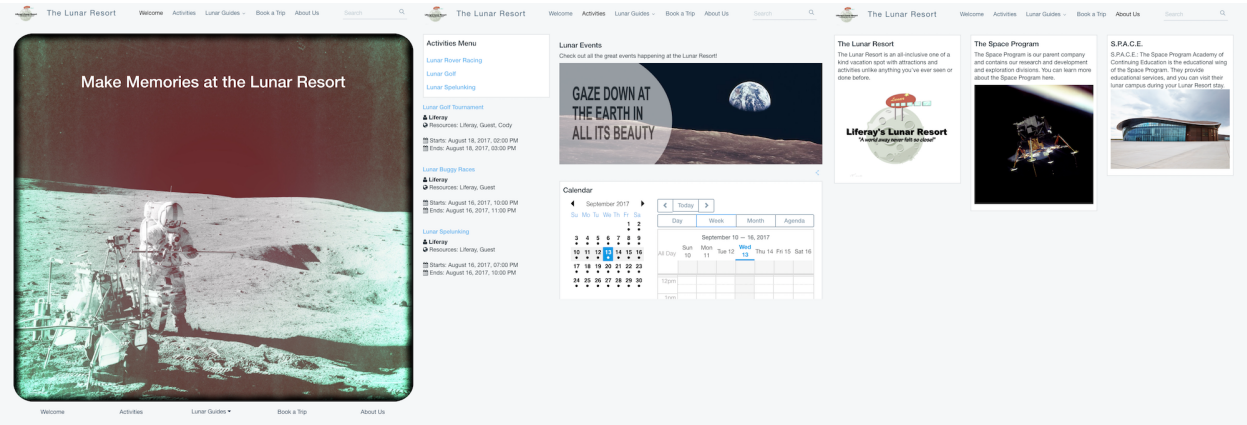


Figure 2.1: A preview of the final site.

You'll start by creating simple content in an empty Liferay DXP bundle. As you progress, you'll add more complex content using Liferay DXP's full suite of tools. These tools include web content display, asset publisher, content display pages, web content structures and templates, page templates, application display templates, and more.

Let's Go!

Chapter 3

Creating Basic Web Content

As soon as Liferay DXP is installed, you have all the tools you need to build sites and create content. This means you can start right away! The first thing to learn is the tool into which your creative juices flow: Liferay DXP's content editor.

To do this, you'll first create a static piece of web content, and then in a second step, you'll publish it. In five or ten minutes, you'll already have a good idea how content creation works in Liferay DXP. What's stopping you?

Download the necessary files to complete the exercises.

Let's Go!

3.1 Initial Setup

```
<p>Creating Basic Web Content<br>Step 1 of 3</p>
```

Before you get any further you should set your site name. When you start a new Liferay bundle for the first time, it creates a site with a single "Welcome" page. The default name for the Site is "Liferay," but you can change that by changing the name globally. You can set the name through the Setup Wizard or through the Control Panel.

Changing the Site Name

Since you're working on the Lunar Resort website, and not Liferay's website, you should head over to the Control Panel to set your site name properly.

1. Open the Menu.
2. Click on *Control Panel → Configuration*
3. Select *Instance Settings*.

On the Instance Settings page, you can configure many aspects of your Liferay Instance, ranging from the site name and logo to authentication and email configuration. All you're concerned about today is the site name.

1. Under *General → Main Configuration* enter *The Lunar Resort* in the *Name* field.

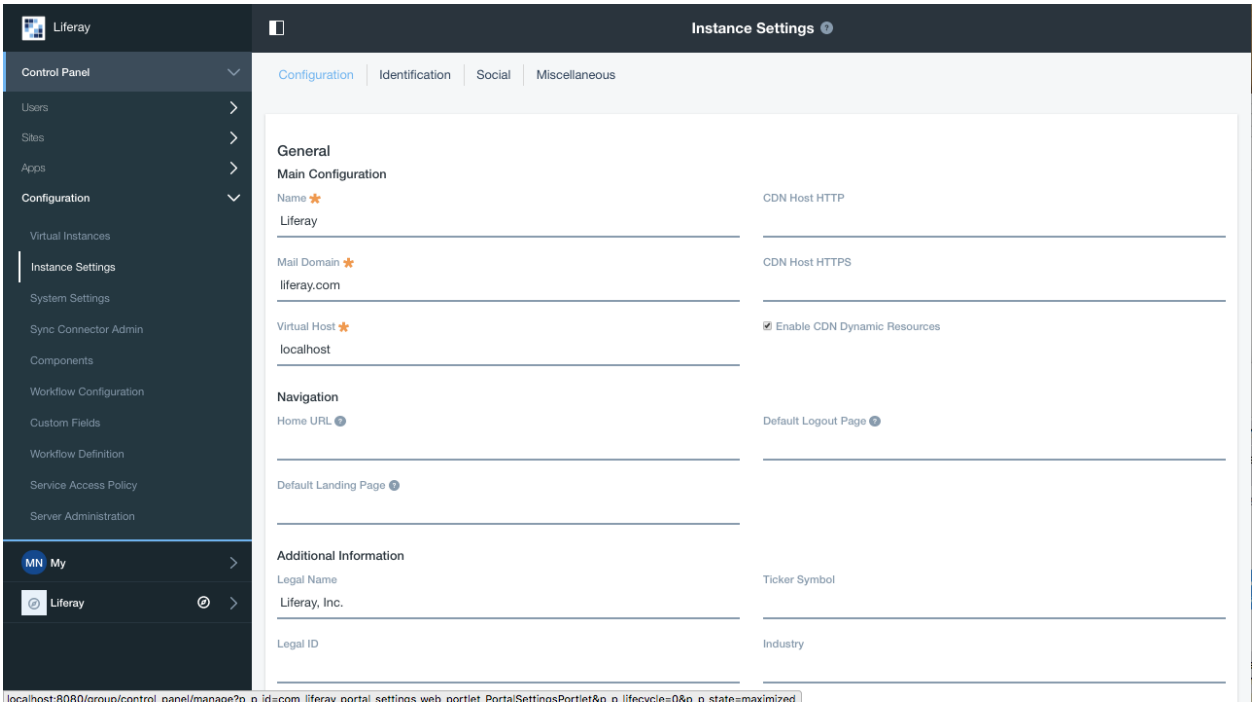


Figure 3.1: The Instance Settings page.

2. Scroll down to the bottom and click *Save*.

3. Instantly, you should see that "The Lunar Resort" has replaced "Liferay" wherever the site name is referenced.

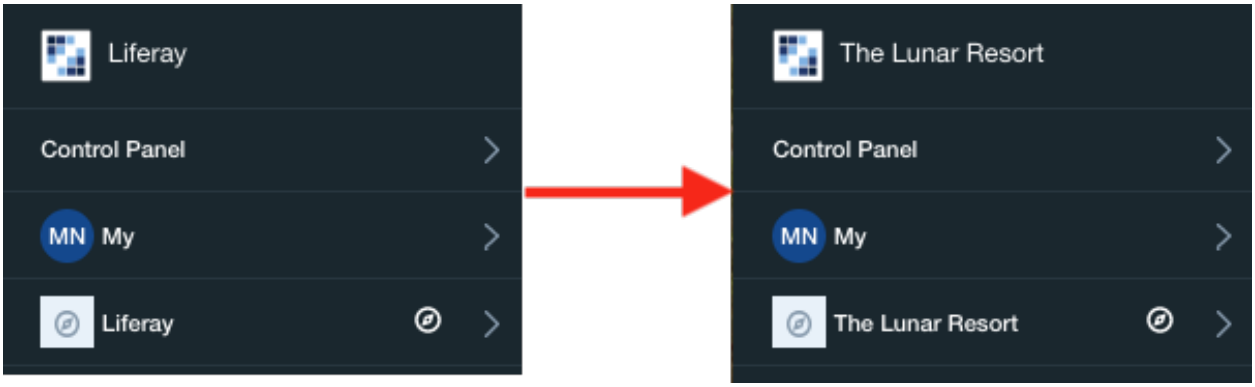


Figure 3.2: The Instance Settings page.

Now that we have that little bit of housekeeping taken care of, you can continue to create content and build your site!

3.2 Using Liferay DXP's Content Editor

Creating Basic Web Content
Step 2 of 3

In Liferay DXP, you create content independent of a page and then publish it on a page. Separating content creation from content display lets you publish content on multiple pages without recreating it, and prevents the loss of content if you remove it from a page or delete the page altogether.

!PVideo Thumbnail

It's time to get started:

1. Open the Menu.
2. Under the site (*The Lunar Resort*), select *Content* → *Web Content*.

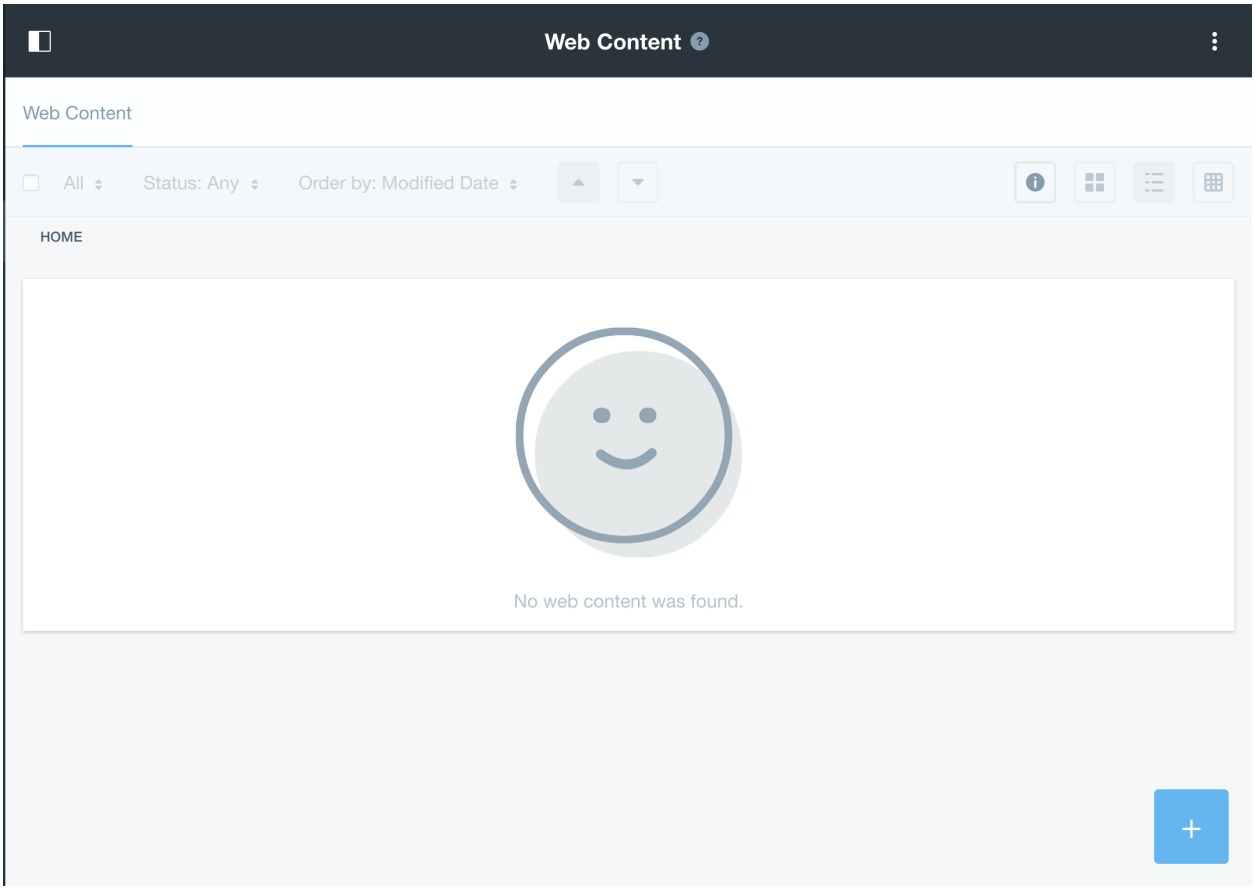


Figure 3.3: The Web Content page.

The Web Content site administration page is the hub for all the content in the Lunar Resort site. You create, edit, and review content from this page.

Follow these steps to create some content:

1. Click the + button in the bottom right corner of the screen and select *Basic Web Content*.

   This opens up the web content editor. There are many options here, but for now pay attention only to the *Title*, *Summary*, and *Content* fields:

- *Title*: The content's title
- *Summary*: The content's summary. This should be a brief description of the content.
- *Content*: The content's body. For example, if the content is an image, you add image in this field.

2. In the Title field, enter *Lunar Resort Welcome*.
3. In the Summary field, enter *Welcome page content for the Lunar Resort*.

The Content field hides a great deal of power. It's sort of like a super hero. Now you'll use it to create your first piece of web content:

1. In the Content field, enter the text *Make Memories at the Lunar Resort*.
2. Select the text with your cursor.
3. In the toolbar that appears above the text, change *Normal* to *Heading 1*. This sets the text as a first-level heading. You may need to close the menu for the full dialog above the text to be visible.

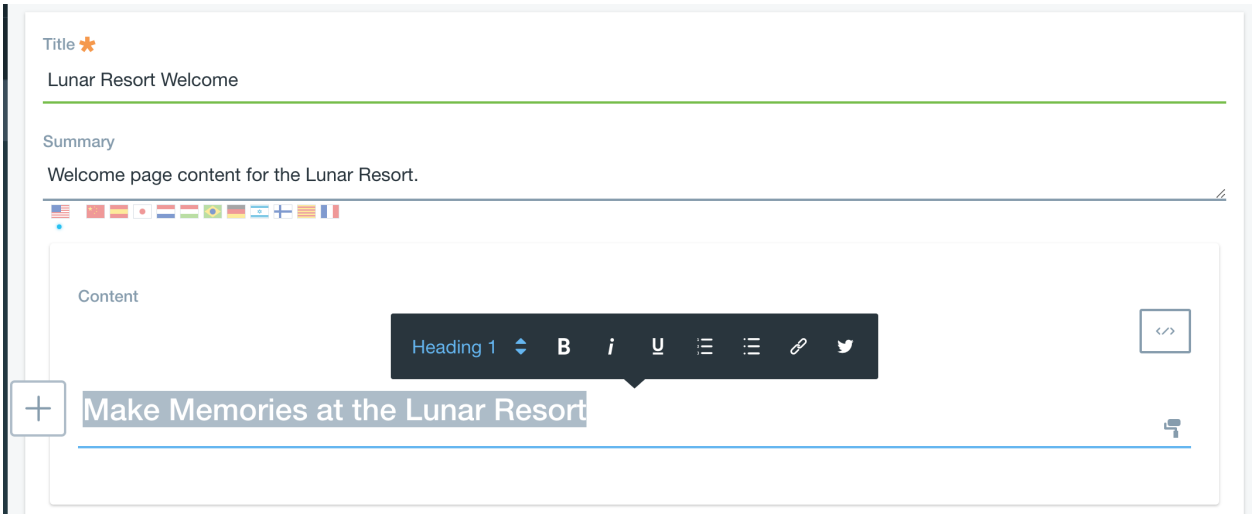


Figure 3.4: The Content field lets you format text on the fly.

4. De-select the text and make a new line.
5. Click the plus button and select the image icon.

Figure 3.5: To begin adding an image, select the image icon.

6. Using drag and drop or the *Select File* button, upload the provided `moon-image1.jpg`. This also adds it to the Document Library so that you can re-use it later.

7. Click *Add* to add the image to the content.

8. If the image must be resized, click it and then drag the corners.

9. Click *Publish*. This saves your web content and makes it available for display.

Well done! You created the content, but you must still publish it on a page so users can see it. You'll do this next.

!VVideo Tutorial

3.3 Publishing Basic Content

Creating Basic Web Content
Step 3 of 3

Before you publish content, it's useful to know the content hierarchy in Liferay DXP. You can loosely think of this as the scaffolding that Liferay DXP uses to display content.

- A Site (like The Lunar Resort) contains pages and content.
- Pages contain applications.
- You can publish a site's content in an application on a page in the site. For example, the Web Content Display and Asset Publisher applications can display a site's content once they're added to a page.

!PVideo Thumbnail

If you followed the previous steps, you have a site that contains content and a page. Now you'll publish that content on that page.

1. Return to The Lunar Resort's Welcome page. If you're still on the Web Content page after creating the web content in the previous step, click *The Lunar Resort* in the top-left corner of the screen.

2. Open the *Add* menu. Select *Content* if it's not already selected.

3. Drag and drop *Lunar Resort Welcome* onto the page above the *Hello World* portlet.

Behind the scenes, Liferay DXP placed your content inside a Web Content Display application and added that application to the page. You could have done this manually by adding a Web Content Display application to the page and then configuring it to display the content. But it's much easier to let Liferay DXP do it for you!

Cleanup

You probably noticed a couple of sub-optimal things about your page: your content's title looks a bit odd, and the Hello World portlet is unnecessary now that the page has content. You'll fix this now.

1. Mouse over your content to reveal the portlet bar.

2. Click the portlet menu icon and select *Look and Feel Configuration*.

3. Under *Application Decorators* select *Barebone*. This removes the content's title, background, and border from the content display. Click *Save*.

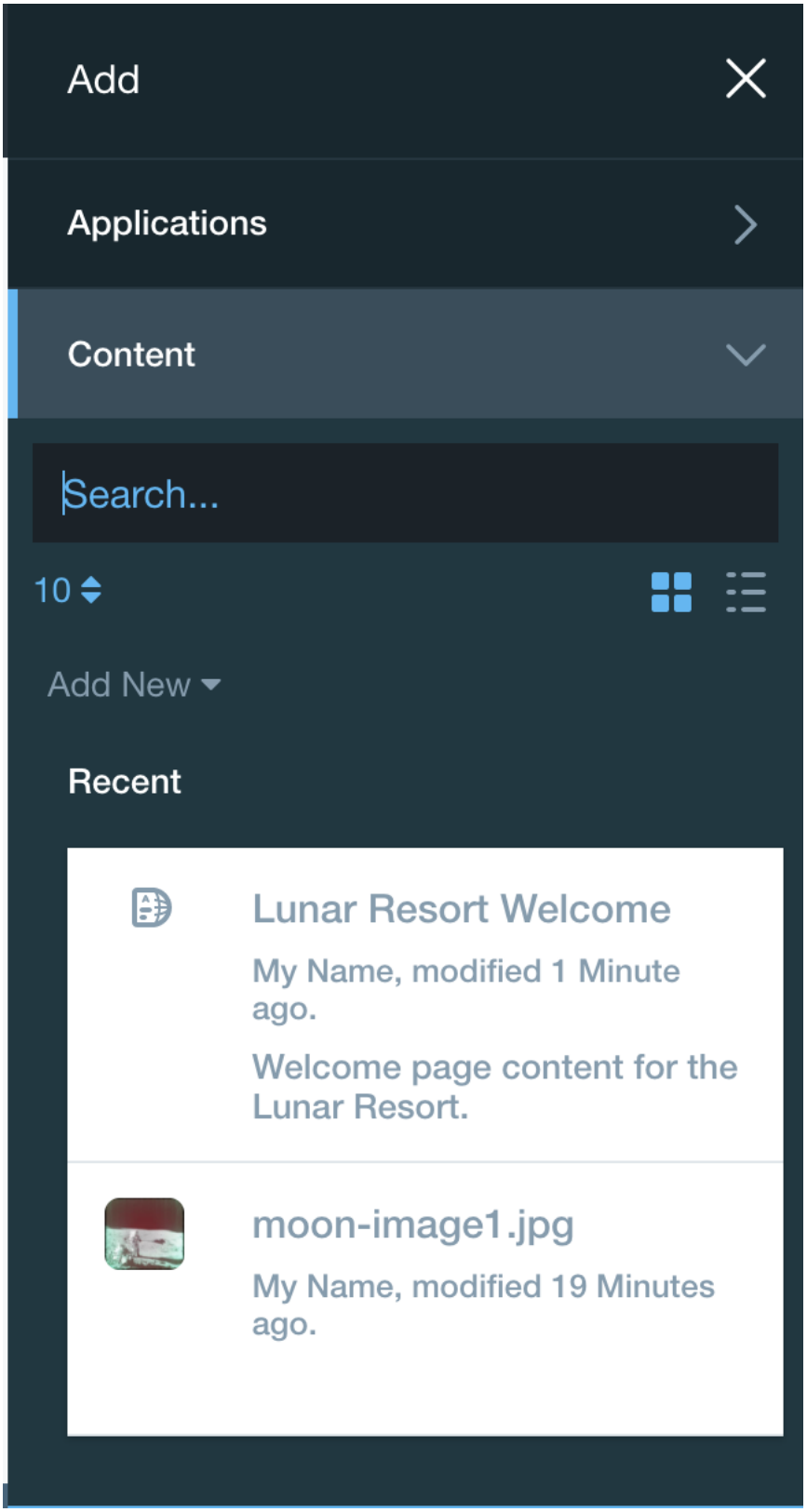


Figure 3.6: The Add menu with your content.

Figure 3.7: Your content on a page.

4. Scroll down to the bottom of the page and mouse over the *Hello World* portlet.
5. Click the portlet menu icon and select *Remove*.
6. Click *OK* when asked if you want to remove the component.

Great work! You've created content, displayed it on a page, and done some basic page management. Now that you know the basics, you'll dig deeper and create some more pages to fill with content.

!VVideo Tutorial

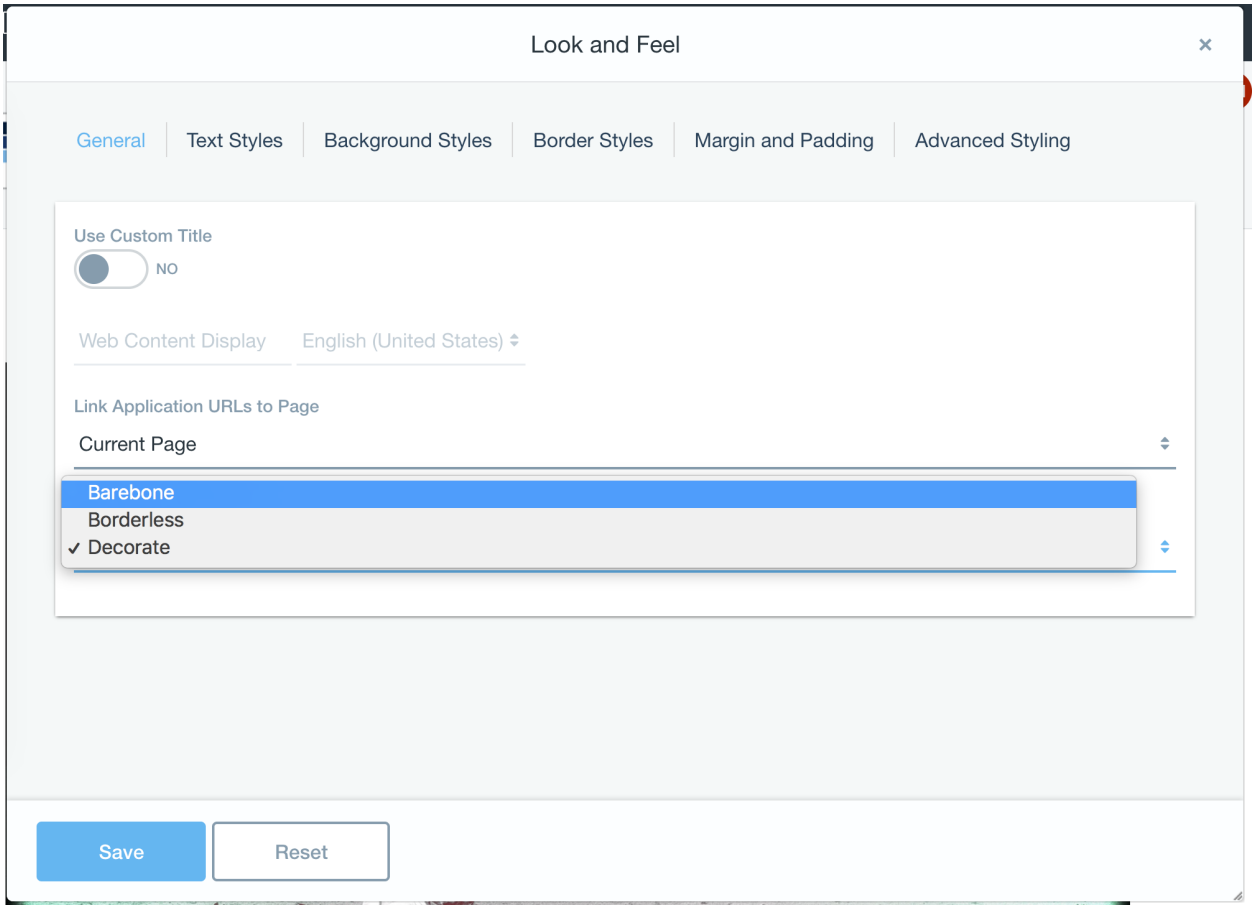


Figure 3.8: Change the portlet's look and feel.

CHAPTER 4

CREATING A SITE

Now that you've created basic web content, you're ready for something a bit more intricate than a single page with a line of text and an image. In this part of the Learning Path, you'll create more pages (including nested pages) and use page layouts to customize how those pages show applications and content. You'll also learn more about adding applications to pages and how to manage site navigation.

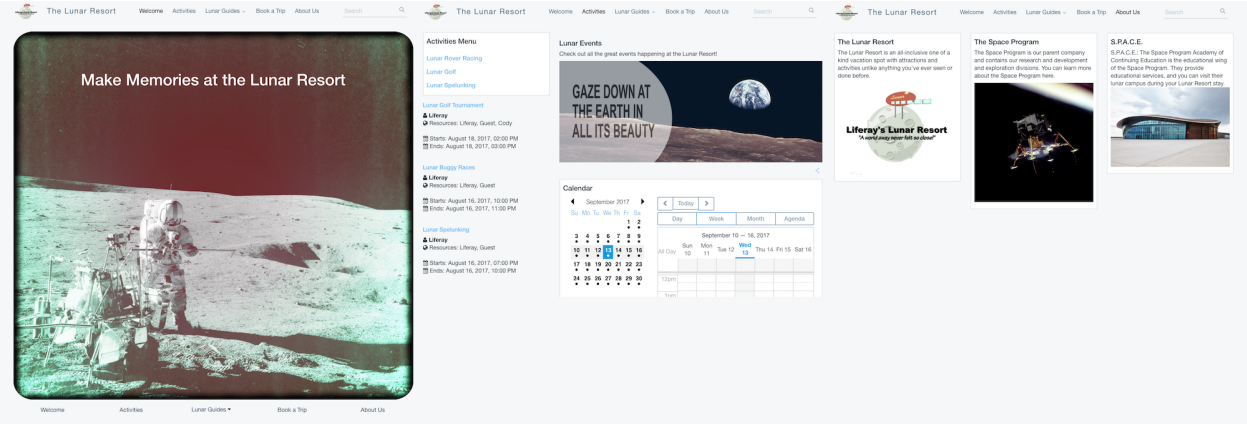


Figure 4.1: A preview of part of the site you'll create.

Let's Go!

4.1 Creating Pages with Layouts

```
<p>Creating A Site<br>Step 1 of 6</p>
```

Up to this point everything on the Welcome page stretches the width of the entire page. You can control this with *page layouts*. For example, the Welcome page has the simplest possible layout: one column. Any items in that column take up its full width. You can place anything above or below but not to the side of any existing item on the page. Liferay DXP contains other layouts that arrange items into different numbers of columns and rows. This gives you a great deal of control over how applications and content appear in your site.

You'll next use layouts to create the Activities and About Us pages for the Lunar Resort.

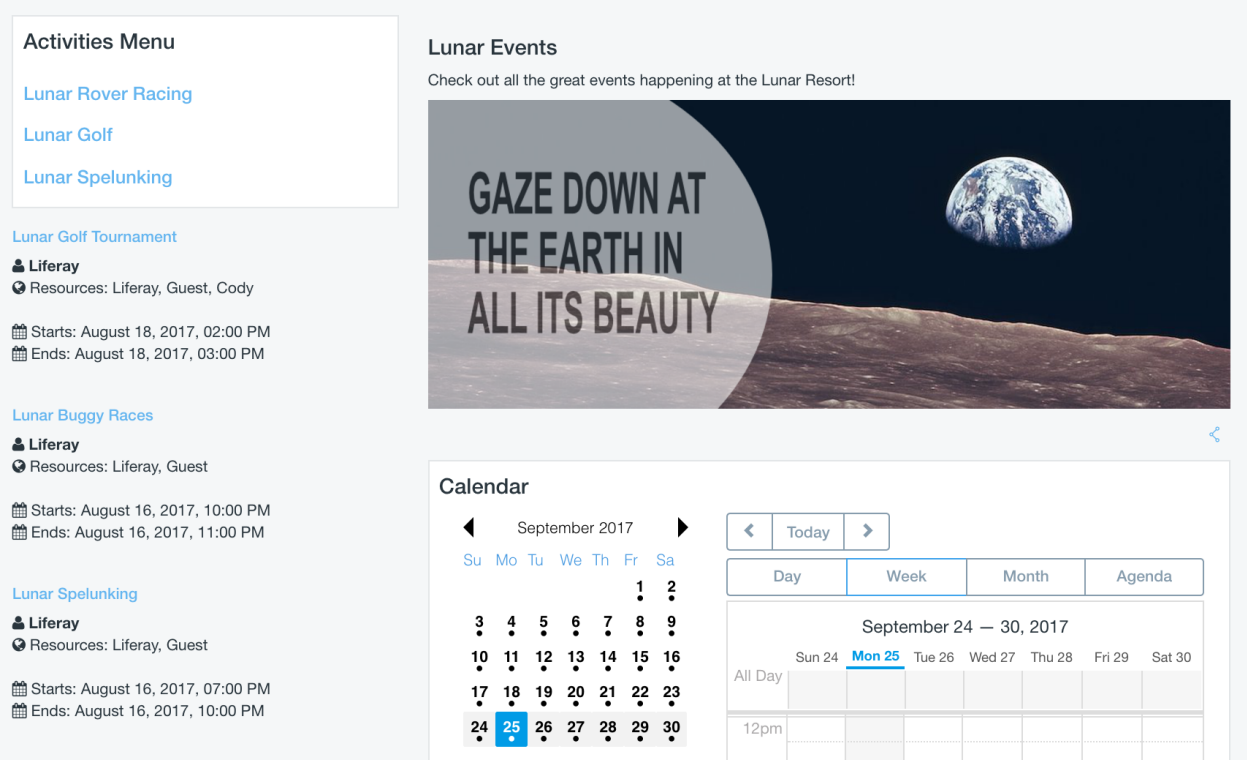


Figure 4.2: The final Activities page.

Creating the Activities Page

As you may be able to tell from the above screenshot, the Activities page uses a two column layout with a small column on the left and a large column on the right. Follow these steps to create this page:

1. Open the main menu and expand the *Navigation* menu for *The Lunar Resort*.

2. In the Navigation menu, click the *Options* button ( ) for *Public Pages* and select *Add Public Page*.

   You are now on the page creation form. You can set the page name, choose a template or layout, and set several other options.

3. Enter *Activities* for the *Name*.

4. For the layout, select *2 Columns (30/70)*. This sets the page to the two column layout. The smaller left column takes up 30% of the page, and the larger right column takes up 70% of the page.

5. Leave the other options on their default settings and click *Add Page*.

The Lunar Resort site now contains the Activities page. This page appears in the navigation, and you can add applications and content to it just as you would any other page.

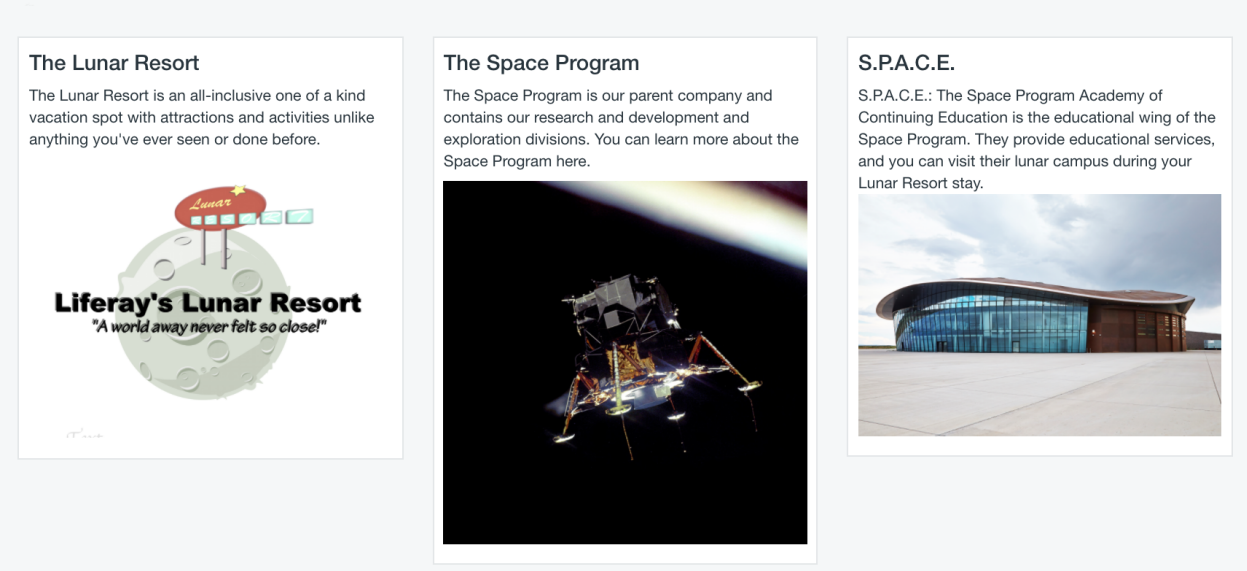


Figure 4.3: The final About Us page.

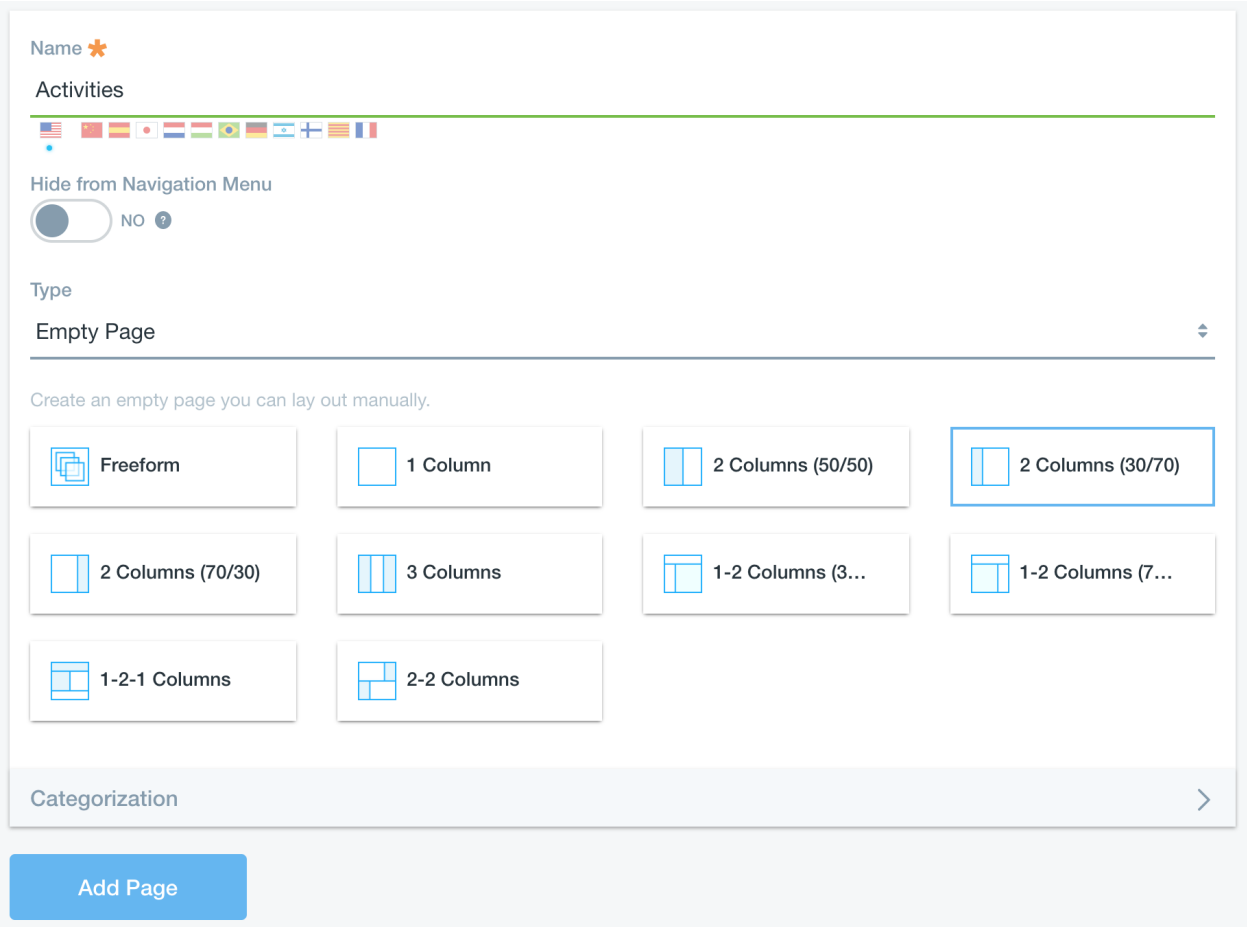


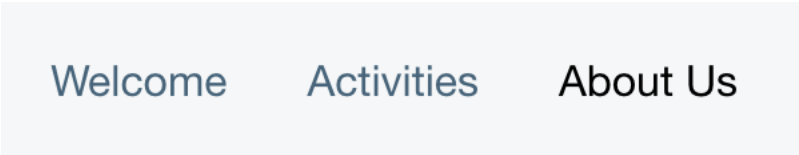
Figure 4.4: Activities page creation.

Creating the About Us Page

Now you'll create the *About Us* page. Note in the above screenshot that this page has three columns. Follow these steps to create this page:

1. In the main menu, expand the *Navigation* menu for *The Lunar Resort*.
2. In the Navigation menu, click the *Options* button ( ) for *Public Pages* and select *Add Public Page*.
3. Enter *About Us* for the Name.
4. For the layout, select *3 Columns*.
5. Leave the other options on their default settings and click *Add Page*.

The navigation bar now contains the two new pages that you just created. Next, you'll create a couple more pages and then arrange their order in the navigation bar.



4.2 Building the Lunar Guides and Book a Trip Pages

```
<p>Creating A Site<br>Step 2 of 6</p>
```

The Lunar Resort needs two more pages:

- Lunar Guides: A page that lists the guides employed by the Lunar Resort.
- Book a Trip: A page for booking a trip to the Lunar Resort.

You'll create these pages using the same steps you used to create the existing pages.

Creating the Lunar Guides Page

1. Open the main menu and expand the *Navigation* menu for *The Lunar Resort*.
2. In the Navigation menu, click the *Options* button ( ) for *Public Pages* and select *Add Public Page*.
3. Enter *Lunar Guides* for the Name.
4. For the layout, select *3 Columns*.
5. Leave the other options on their default settings and click *Add Page*.

Creating the Book a Trip Page

1. Open the main menu and expand the *Navigation* menu for *The Lunar Resort*.
2. In the Navigation menu, click the *Options* button ( ) for Public Pages and select *Add Public Page*.
3. Enter *Book a Trip* for the Name.
4. For the layout, select *1 Column*.
5. Leave the other options on their default settings and click *Add Page*.

Arranging Pages

The new pages now appear in the navigation bar with the other pages. If these pages are out of your preferred order, you can rearrange them via drag and drop. Rearrange the pages to match this order:

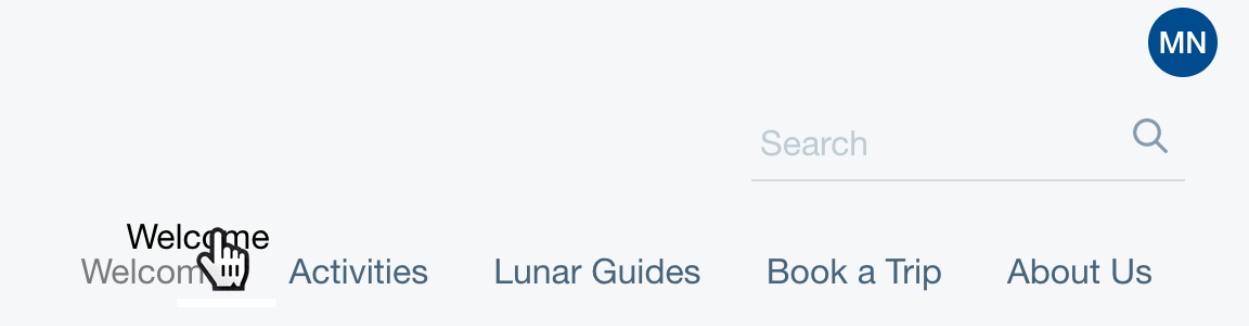


Figure 4.5: Reorder the pages in the navigation bar.

Great! Now that you have your site's pages, you'll add some applications to them.

4.3 Adding Applications to Pages

```
<p>Creating A Site<br>Step 3 of 6</p>
```

All of a page's functionality comes from applications. This functionality can be as simple as displaying content, or as complex as social networking. You create your page's functionality by adding applications to the page. You'll get started by adding applications to the Activities and About Us pages.

The Activities Page

You'll add two Asset Publisher applications to display different assets, like images, web content articles, or any other kind of content in Liferay DXP. Follow these steps to add two Asset Publishers to the page:

1. Navigate to the *Activities* page.
2. Click the *Add* button ( ) on the upper right and expand the *Applications* → *Content Management* category in the menu.
3. Drag and drop one Asset Publisher to the page's right column and the other to the page's left column.

The About Us Page

To add functionality to the About Us page, you'll add several Web Content Display applications. Recall that this application displays web content. You saw it in action when you displayed basic web content in the previous section of this Learning Path. Follow these steps to add Web Content Display to the About Us page:

1. Navigate to the *About Us* page.
2. Click the *Add* button ( ) on the upper right and expand the *Applications* category in the menu.

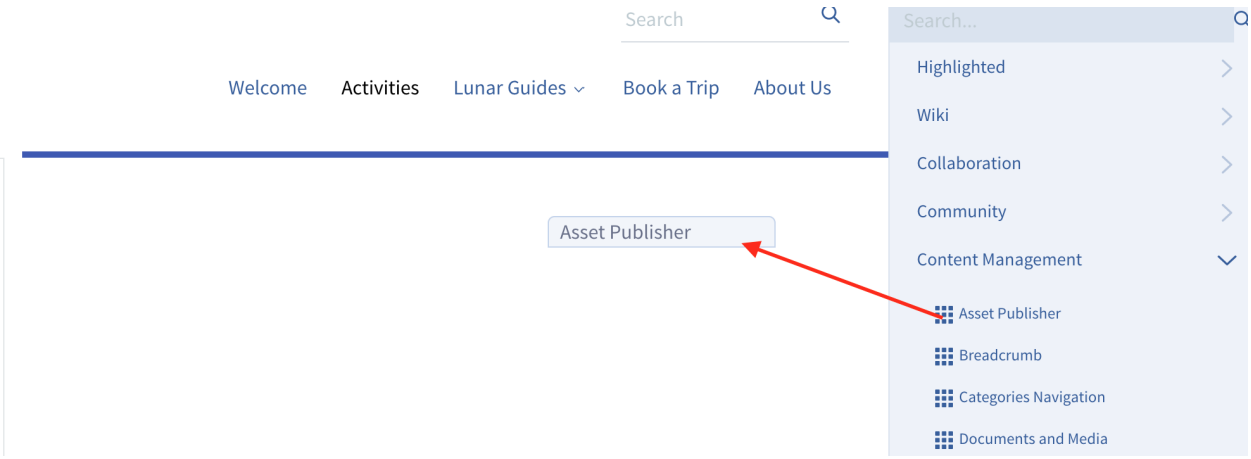


Figure 4.6: This screenshot shows the Asset Publisher being placed in the page's right column. The narrow blue bar indicates where the application will appear when you release the mouse button.

3. This time use the search bar to search for *web content display*.
4. This page's layout has three columns. Drag and drop a Web Content Display application into each column.

As you can see, adding content and applications to pages is a matter of drag and drop. Next, you'll learn about page management and navigation.

4.4 Page Templates

```
<p>Creating A Site<br>Step 4 of 6</p>
```

As you've now seen, creating pages can be repetitive. Wouldn't it be great if you could create multiple pages from a single template? Well guess what: page templates in Liferay DXP let you do exactly that! They also inherit future changes to those pages.

Now you'll use page templates in The Lunar Resort. First you'll create a page template, then you'll use it to create several pages. Later, you'll make changes to the template and see Liferay DXP propagate those changes to the pages you created from the template.

Creating a Page Template

Use these steps to create a page template:

1. Open the Main Menu and select *Control Panel* → *Sites* → *Page Templates*. The Page Templates page lists all the page templates in the Liferay DXP instance. Three page templates come bundled with Liferay DXP. *Blog* and *Wiki* are example layouts for the Blogs and Wiki applications. *Content Display Page* serves a special function that you'll work with later.
2. Click the *Add* button ( ) in the lower right corner. This takes you to the page template creation page.
3. Name the template *Lunar Guide Page*.
4. For the description, enter *Page with information about a Lunar Guide*.

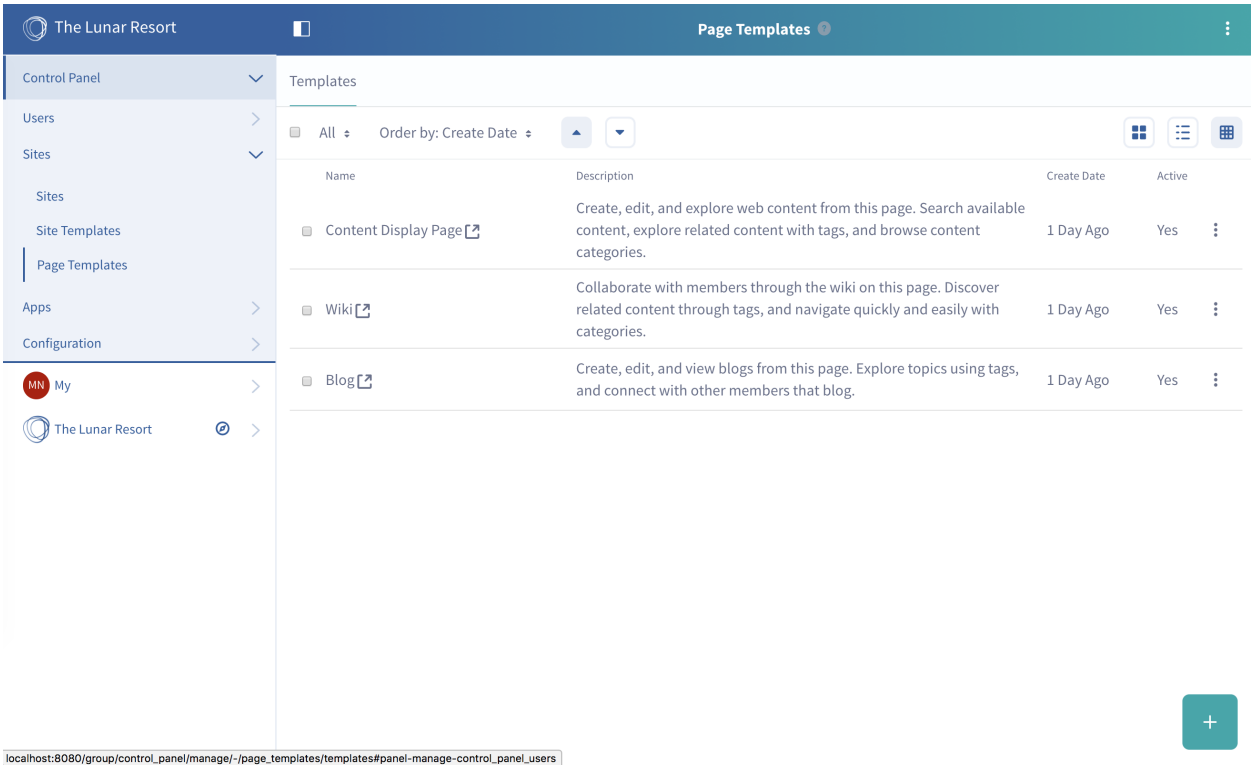


Figure 4.7: The Page Templates page.

5. Click *Save*. This takes you back to the list of page templates, where you can see the new template in the list.

Editing a Page Template

Editing a page template is similar to editing any page. You drag and drop applications onto the page and reposition or remove them as desired. The only difference is that you can't directly add content, and some configuration and display options are disabled (it's a template, after all). Follow these steps to edit the page template you just created:

1. In the list of page templates, click *Lunar Guide Page*. This opens it in a new browser tab or window.
2. In the page template's edit toolbar click the *Add* button ( ) on the upper right and expand the *Applications* → *Collaboration* category. Add a Blogs portlet to the page template's right column.
3. Close the page template's edit tab/window. Liferay DXP automatically saves your changes.

Next, you'll create a page with this template.

Creating a Page with a Page Template

Follow these steps to use the template to create a page:

1. Open the Main Menu and select *The Lunar Resort* → *Navigation*.

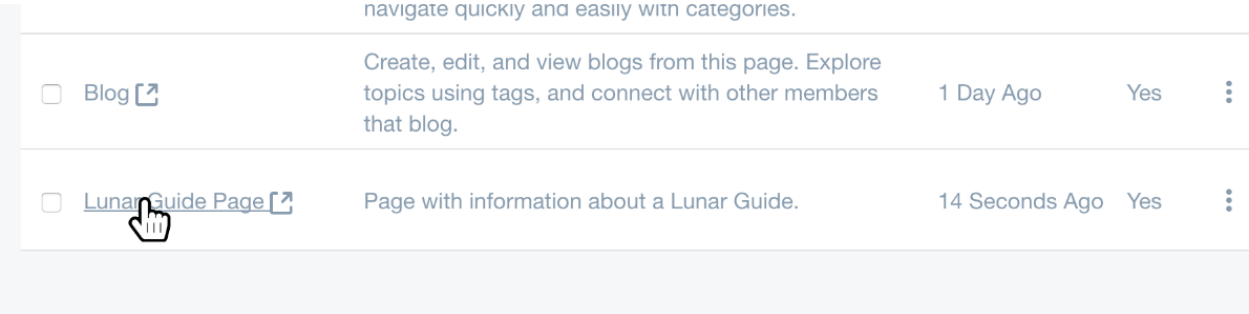


Figure 4.8: Click the page template to edit it.

2. In the Navigation menu, click the *Options* button ( ) for *Public Pages* and select *Add Public Page*.
3. Name the page *Cody* (Cody is one of our lunar guides).
4. Under *Type*, select *Lunar Guide Page*. This selects your page template as the source for the page.
5. Leave *Inherit Changes* set to *Yes*. This lets you edit this page in the future by editing the template, but removes the ability to edit the page directly.
6. Click *Add Page*.

Liferay DXP then creates the new page from your template. Next, you'll create more of these pages, one for each lunar guide (Cody is awesome, but he can't do everything by himself). Note, however, that Cody's page is in the navigation bar next to all the other pages. To get his page and those of the other lunar guides under the Lunar Guides page, you must nest the pages. You'll learn how to do this next.

4.5 Nesting Pages

```
<p>Creating A Site<br>Step 5 of 6</p>
```

When creating sites, you'll likely encounter situations where you want to nest pages under other pages. Such child pages (also called nested pages) let you create page hierarchies to organize content and functionality. For example, the pages for each lunar guide should be nested under the Lunar Guides page. Although this is a simple use case, note that pages in Liferay DXP can be nested to unlimited levels. This lets your site support even the most demanding hierarchies, so long as you can design a UI to handle it.

In this article, you'll nest the existing lunar guide page for Cody under the Lunar Guides page. You'll then create more lunar guide pages as child pages of the same Lunar Guides page.

Creating Child Pages

There are two ways to create a child page in Liferay DXP:

1. Create a new page as a child page of an existing page. This is the most common way.
2. Turn an existing page into a child page of another existing page. You can even do this via drag and drop.

You'll start with the second option, since you already created a page (Cody) that you want to nest under the Lunar Guides page.

Creating Child Pages with Drag and Drop Using a site's *Navigation* menu, you can nest pages via drag and drop. Liferay DXP immediately applies any changes you make here to the site's navigation structure. Follow these steps to nest the *Cody* page under the *Lunar Guides* page:

1. Open the Main Menu and select *Lunar Resort* → *Navigation*.
2. Drag and drop the page *Cody* and so that it nests under the *Lunar Guides* page.

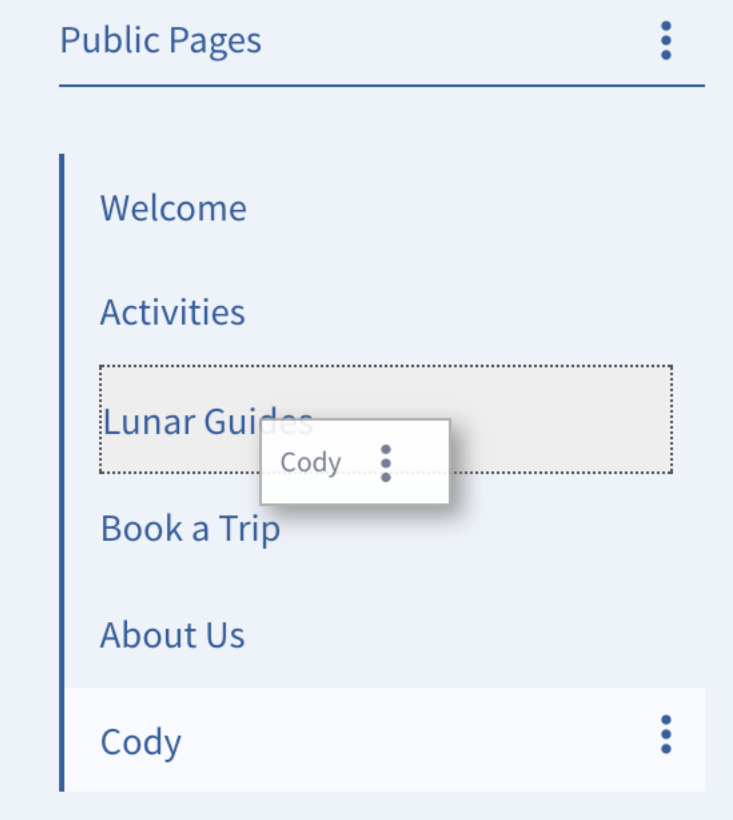


Figure 4.9: Nesting a page with drag and drop.

3. Refresh the page. The new page hierarchy now appears in the navigation bar.

Nice work! Next, you'll create the rest of the pages for the lunar guides as child pages of the *Lunar Guides* page. Cody is about to have some company.

Creating New Child Pages The Lunar Resort's other lunar guides–Jim, Steve, and Russ–also need pages nested under the *Lunar Guides* page. You'll create these pages directly as child pages:

1. In the *Lunar Resort* → *Navigation* menu, click the *Options* button ( ) for *Lunar Guides* and select *Add Child Page*.
2. Name the page *Jim* and set its type as *Lunar Guide Page*.
3. Leave *Inherit Changes* set to *YES*, and click *Add Page*.
4. Repeat these steps to create pages for Steve and Russ.

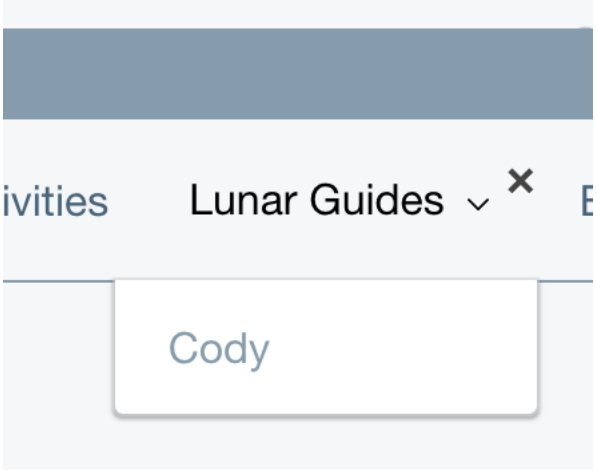


Figure 4.10: The page *Cody* is now nested under *Lunar Guides* page.

Liferay DXP creates each page from the template, with the Blogs app in the right column and an empty space in the left column. Each page also appears in the navigation bar under the *Lunar Guides* page.

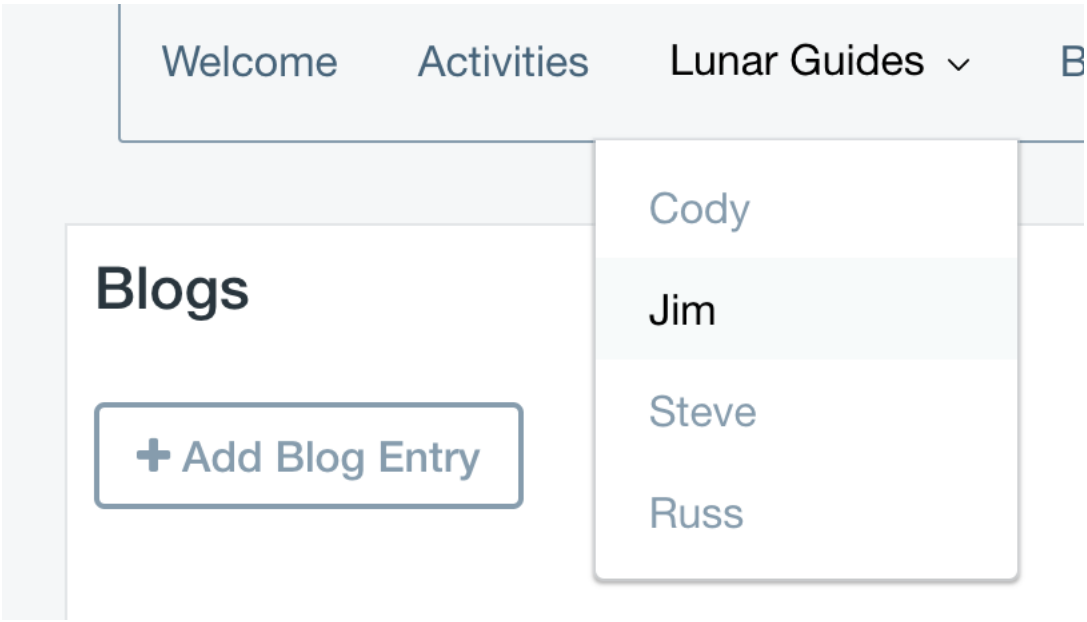


Figure 4.11: Cody is no longer lonely!

Now that you've created all the pages, you'll learn more about site navigation and the various features of Liferay's Breadcrumb and Navigation apps.

4.6 Site Navigation

Creating A Site
Step 6 of 6

You could have the greatest site in the multi-verse, but if users can't navigate it, it's all for naught. Fortunately Liferay DXP provides extensive support for customizing your site's navigation. There are two

main ways to define and customize site navigation:

1. In the site's theme. This is the primary and most powerful way to manage site navigation. Themes in Liferay DXP can customize any aspect of a site, including its navigation. Defining site navigation in the theme provides a uniform look and feel across the site.

2. With apps. Navigation apps can define navigation on a page-by-page basis. For example, the Navigation Menu app displays a navigable page hierarchy of the site. You can select which page in the hierarchy to use as the root. There's also the Breadcrumb app, which displays the trail of pages in the hierarchy that lead to the current page. Like a trail of breadcrumbs in the woods, this app lets users retrace their steps so they don't get lost.

These two ways of defining site navigation can be used together. Like page templates, themes can define which apps appear in a site. You've already seen an example of this in action without being aware of it. The Lunar Resort uses the Liferay DXP's default theme, Classic. This theme embeds the Navigation Menu app as the site navigation bar. Therefore, the Lunar Resort's navigation bar is nothing more than a Navigation Menu app configured to display the entire site's page hierarchy.

You won't change the theme the Lunar Resort uses, but you'll fine-tune the site's navigation by adding another Navigation Menu app and a Breadcrumb app.

Adding a Navigation Menu App

The current Navigation Menu app does a fine job of defining the Lunar Resort's navigation bar, so why would you want to add another one? For footer navigation, of course! Once a page requires scrolling, the navigation bar disappears. It's therefore useful to have another one at the bottom of the page. Follow these steps to add and configure such a Navigation Menu app:

1. Go to the Lunar Resort's *Welcome* page.

2. Click the *Add* button ( ) on the upper right and expand the *Applications* → *Content Management* category in the menu.

3. Drag a *Navigation Menu* app onto the page below the existing content.

4. Mouse over the Navigation Menu app and click the app's *Options* menu ( ) on the right hand side of the portlet bar. Then select *Look and Feel Configuration*.

5. Set *Application Decorators* to *Barebone*, click *Save*, then close the dialog box.

6. Now select *Configuration* from the Navigation Menu app's *Options* menu ( ).

7. For *Display Template*, select *Bar minimally justified styled*.

8. Leave the rest of the default settings alone, then click *Save* and close the dialog box.

The navigation footer should look like this:
Great! Next, you'll add a Breadcrumb app.

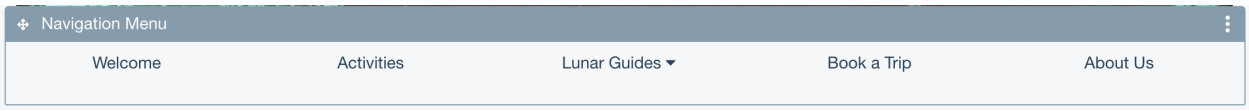


Figure 4.12: The Welcome page now has a navigation footer.

Adding a Breadcrumb App

To add the Breadcrumb app, you'll leverage the power of Page Templates. Recall that you created the Lunar Guide pages using a template with the Blogs app in the right hand column of the page. Since Liferay DXP propagates template changes to the pages, you'll now change the template to add a Breadcrumb app to each of the Lunar Guide pages. Follow these steps to do so:

1. Open the Main Menu and select *Control Panel* → *Sites* → *Page Templates*.
2. In the list of page templates, click *Lunar Guide Page*.
3. Add a *Breadcrumb* application from the Content Management category to the page template's left column (next to the Blogs application).
4. Open Breadcrumb app's *Options* menu ( ) and select *Look and Feel Configuration*.
5. Change *Application Decorators* to *Barebone*, then click *Save* and close the dialog.
6. Close the page template editing tab/window.
7. Navigate to one of the lunar guide pages. The Breadcrumb now appears alongside the Blogs app.

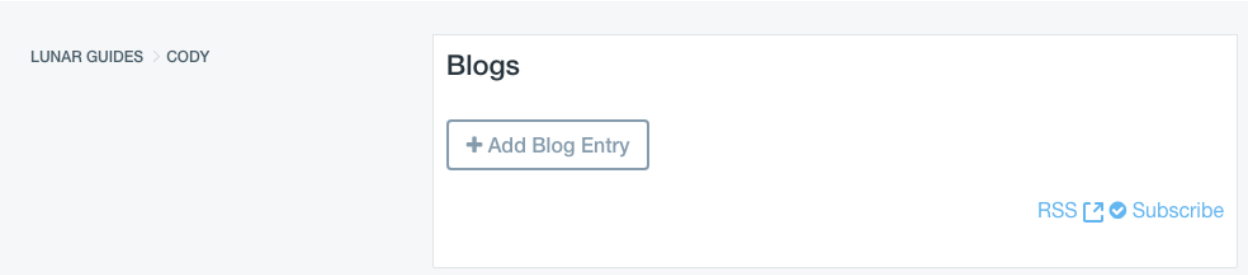


Figure 4.13: After adding the Breadcrumb app to the page template, the app appears on each lunar guide page, to the left of the Blogs app.

As you've seen, the Navigation Menu and Breadcrumb apps help users traverse page hierarchies. These apps are invaluable for any site with complex page hierarchies.

Awesome! Now that you've created your site's framework with pages and navigation, you'll fill the site with content.

Chapter 5

Creating Content

So far you've created the Lunar Resort site's outline. By creating pages and using templates, the site now has a structure and the pages have layouts. Now it's time to color everything in with content.

In this section you'll create content, some with more complex web content structures and templates, and use Liferay's Web Experience Management to publish your new content on your site.

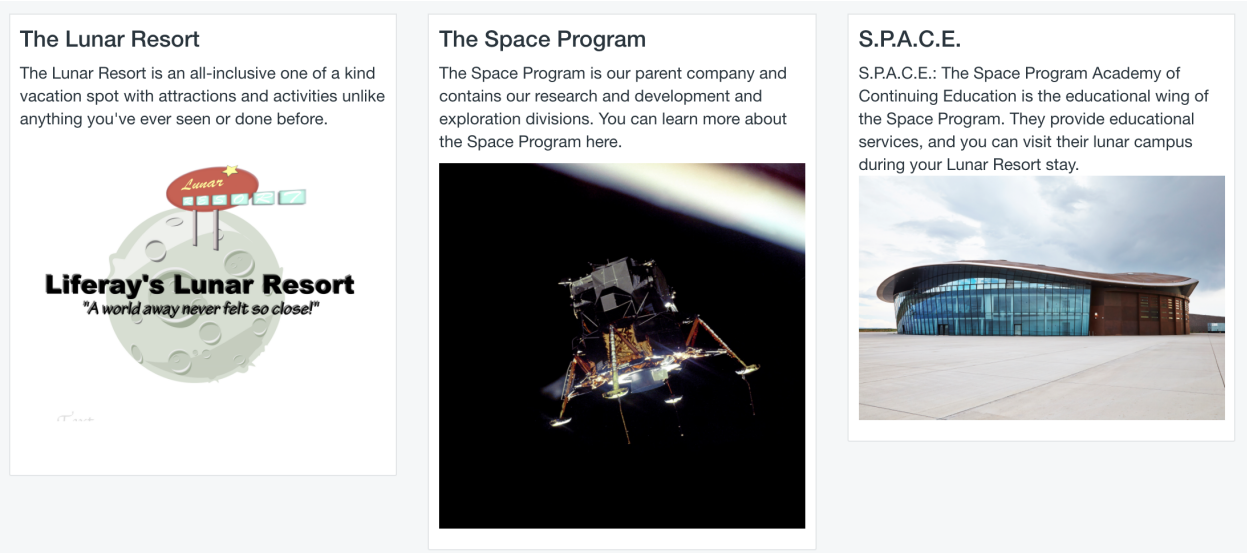


Figure 5.1: Basic web content.

Let's Go!

5.1 Creating More Content

```
<p>Creating Content<br>Step 1 of 7</p>
```

Earlier we created some content without knowing where it would go. Now you have the context of how the site is organized. Next, use the same principles for creating content on the front page to create articles for the *About Us* page.

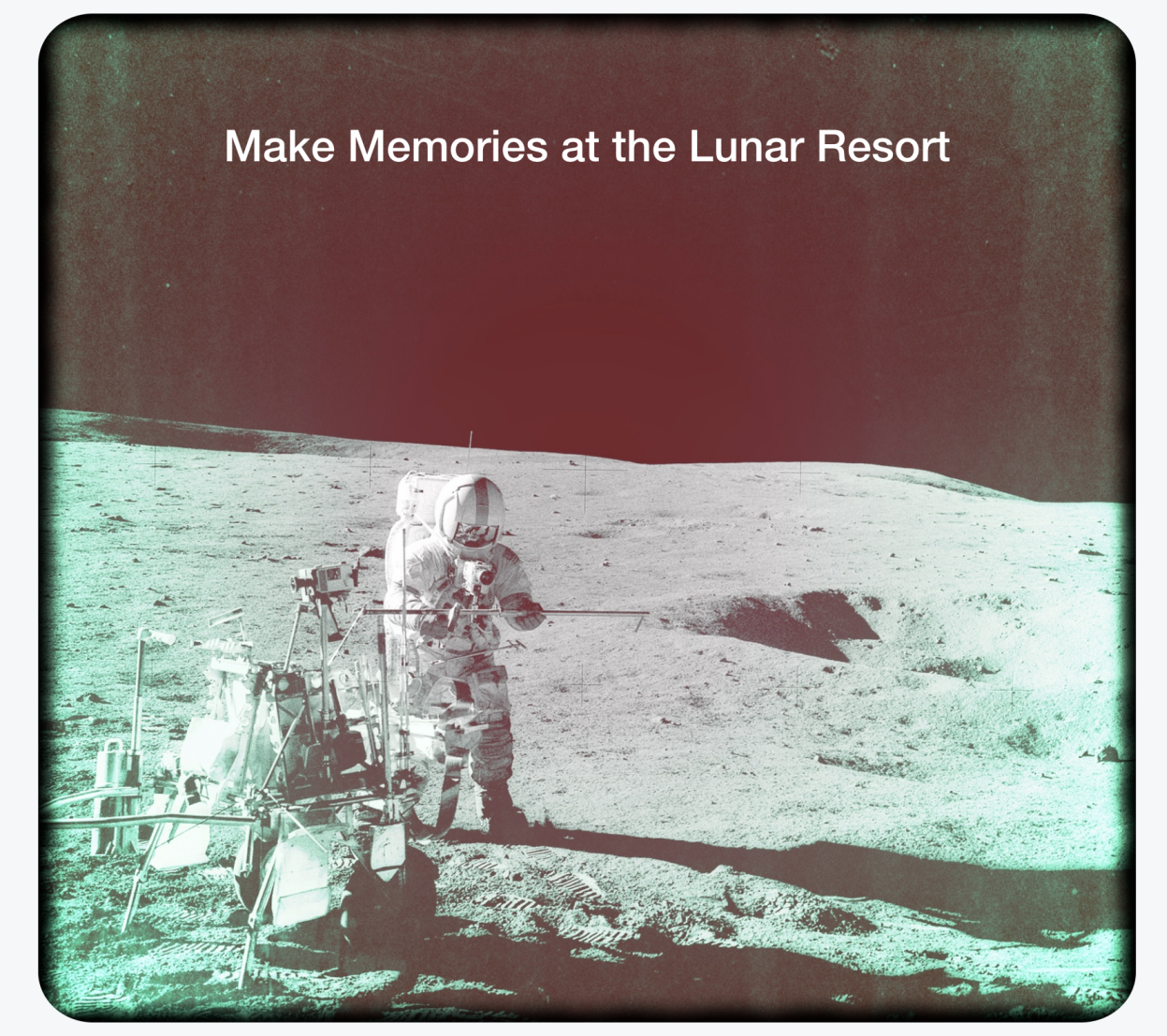


Figure 5.2: Application Display Template.

On the *About Us* page, you'll add information about the Lunar Resort and other related Space Program initiatives.

Creating the About Us Page Content

Recall that you created the Welcome page's content in Site Administration and then added it to the page. You'll use a slightly different method to create the About Us page's content. Specifically, you'll use the Web Content Display app to create the content directly in that app.

1. Go to the *About Us* page.
2. In the portlet bar of the left column's Web Content Display app, click the *Add* button (+) and select *Basic Web Content*. This takes you to the app's web content creation page. The content you create here only appears in this app, on this page.

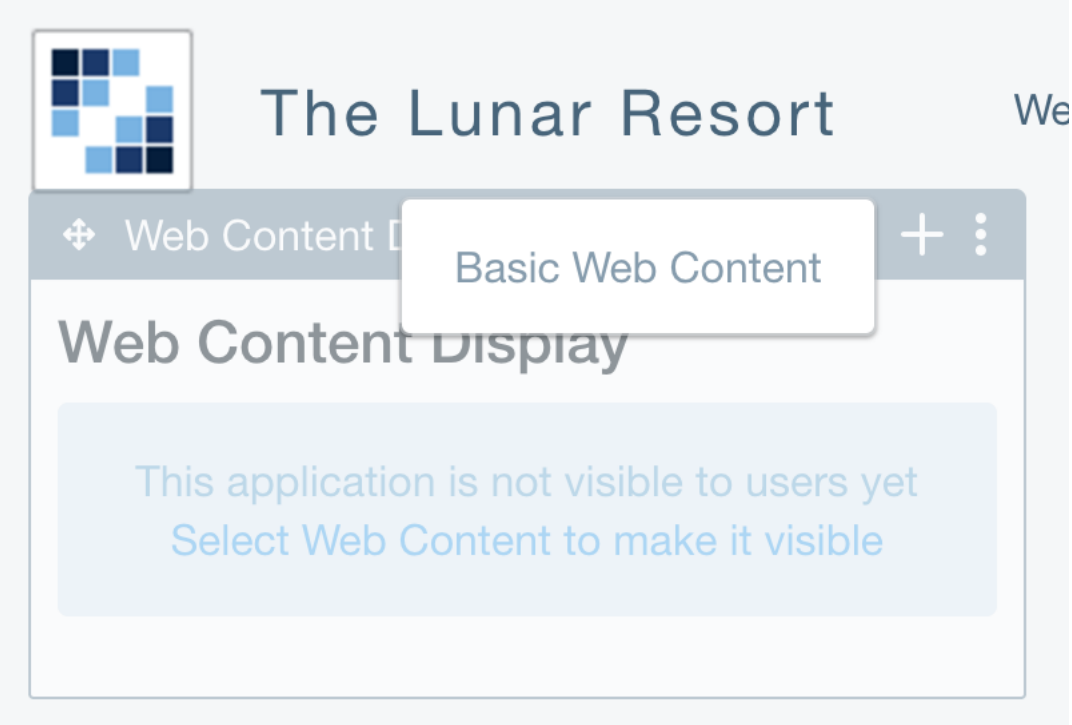


Figure 5.3: You can create basic web content directly in the Web Content Display app.

3. Enter the following information for the following fields:

    - **Title:** The Lunar Resort
    - **Summary:** Information about the Lunar Resort
    - **Content**: The Lunar Resort is an all-inclusive, one of a kind vacation spot with attractions and activities unlike anything you've ever seen or done before.

4. With your cursor still in the Content field one line below the text you just entered, click the plus button and select the image icon. Select `lunar-resort-logo.png`.

5. Resize the image if necessary, then click *Publish*.

The Web Content Display app in the left column now contains your content.

To add content in the remaining Web Content Display apps, repeat the above steps for each app but add different information in step two, and a different image in step three.

1. Add this for the Web Content Display app in the middle column:

    - **Title:** The Space Program
    - **Summary:** Information about the Space Program
    - **Content:** The Space Program is our parent company and contains our research, development, and exploration divisions. You can learn more about the Space Program here.
    - Add the image `space-program-logo.png`.

2. Add this for the Web Content Display app in the right column:

- **Title:** S.P.A.C.E.
- **Summary:** Information about S.P.A.C.E.
- **Content:** The Space Program Academy of Continuing Education (S.P.A.C.E.) is the Space Program's educational wing. You can visit its lunar campus during your Lunar Resort stay.
- Add the image `space-logo.png`.

Great! The About Us page is now complete and provides all the information that the Lunar Resort's visitors need.

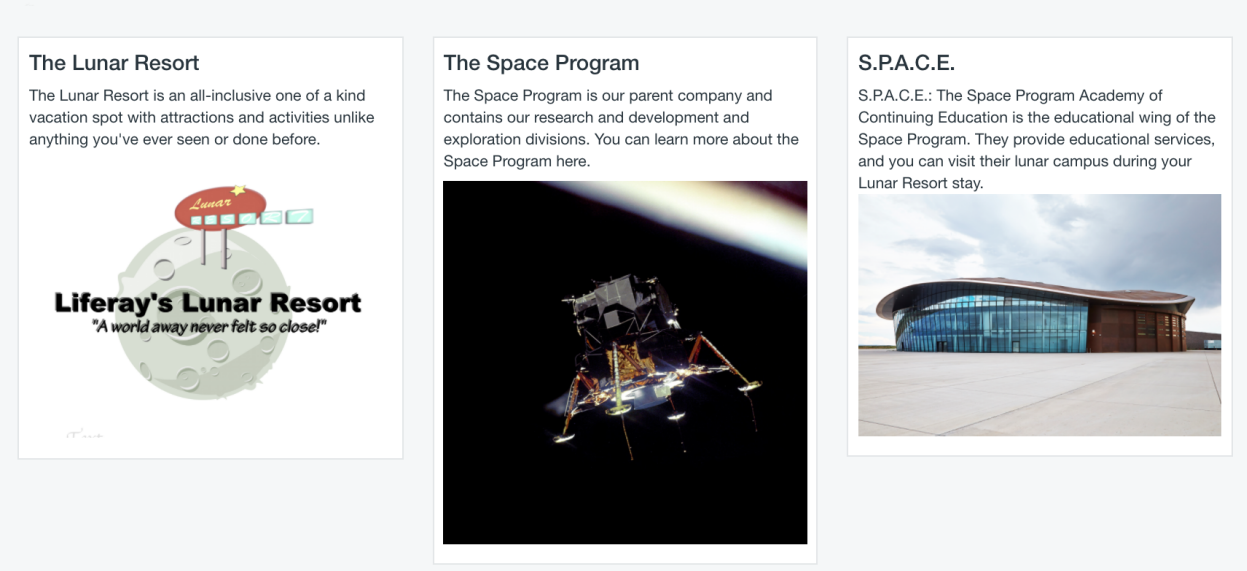


Figure 5.4: The complete About Us page looks awesome!

Next, you'll learn how to use Liferay DXP's Documents and Media features to manage files in your portal.

5.2 Using Documents and Media

```
<p>Creating Content<br>Step 2 of 7</p>
```

Liferay DXP's Documents and Media features provide tools for uploading, organizing, and displaying various types of documents and media, including images, audio, and video. For example, you can use Documents and Media for collaborating on files like text documents or spreadsheets, managing collections of images, or simply storing the images that your web content references.

You've been using Documents and Media without knowing it. The images you added when creating web content were automatically added to the Documents and Media library. But there's a more efficient way to add images to Documents and Media. You can add many images at once, directly in the Documents and Media library. This saves you a step when creating content that uses those images. Working directly in the Documents and Media library also lets you organize your files in folders just as you would on a traditional desktop environment.

Now you'll take advantage of these features in the Lunar Resort.

Adding Files

You still need to create more content in the Lunar Resort, and this content needs images. You'll upload all these images now in the Documents and Media library:

1. Open the *Menu* ( ) and select *The Lunar Resort* → *Content* → *Documents and Media*. The Documents and Media screen appears and displays the Documents and Media library's Home folder (its root folder). Note that the images you added while creating content earlier are all here.

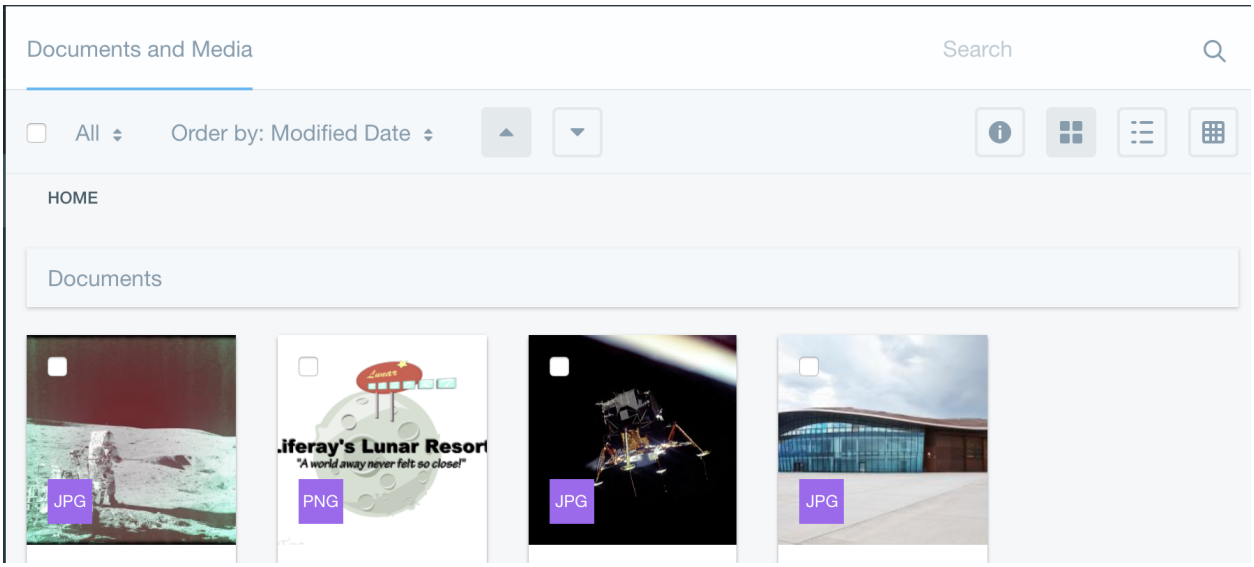


Figure 5.5: The Documents and Media library's Home folder contains the Lunar Resort's existing images.

2. Click the *Add* icon ( ) at the bottom-right of the page and select *Multiple Documents*.
3. Use the *Add Multiple Documents* screen to add the images `moon-image1.jpg`, `moon-image2.jpg`, `moon-image3.jpg`, `booking-image.png`, `guide-cody.png`, `guide-jim.png`, `guide-russ.png`, and `guide-steve.png`. You can either drag and drop the files from your file system to the box, or click *Select Files* to select the files via a file manager window. If you wish, you can also categorize and describe the files. Click *Publish* to complete the upload.

Great! The Lunar Resort's Documents and Media library now contains more images–enough to warrant organizing them in folders.

Adding Folders

Your Documents and Media library is now a bit more crowded. It's time to organize it with folders. Besides basic file organization, you can also use folders for managing access to files via permissions.

Follow these steps to add some folders to the Lunar Resort's Documents and Media library:

1. Return to the Documents and Media library's Home folder.
2. Click the *Add* icon ( ) at the bottom-right of the page and select *Folder*.
3. Name the folder *Web Content Images*, and give it the description *Images used in Lunar Resort Web Content*.

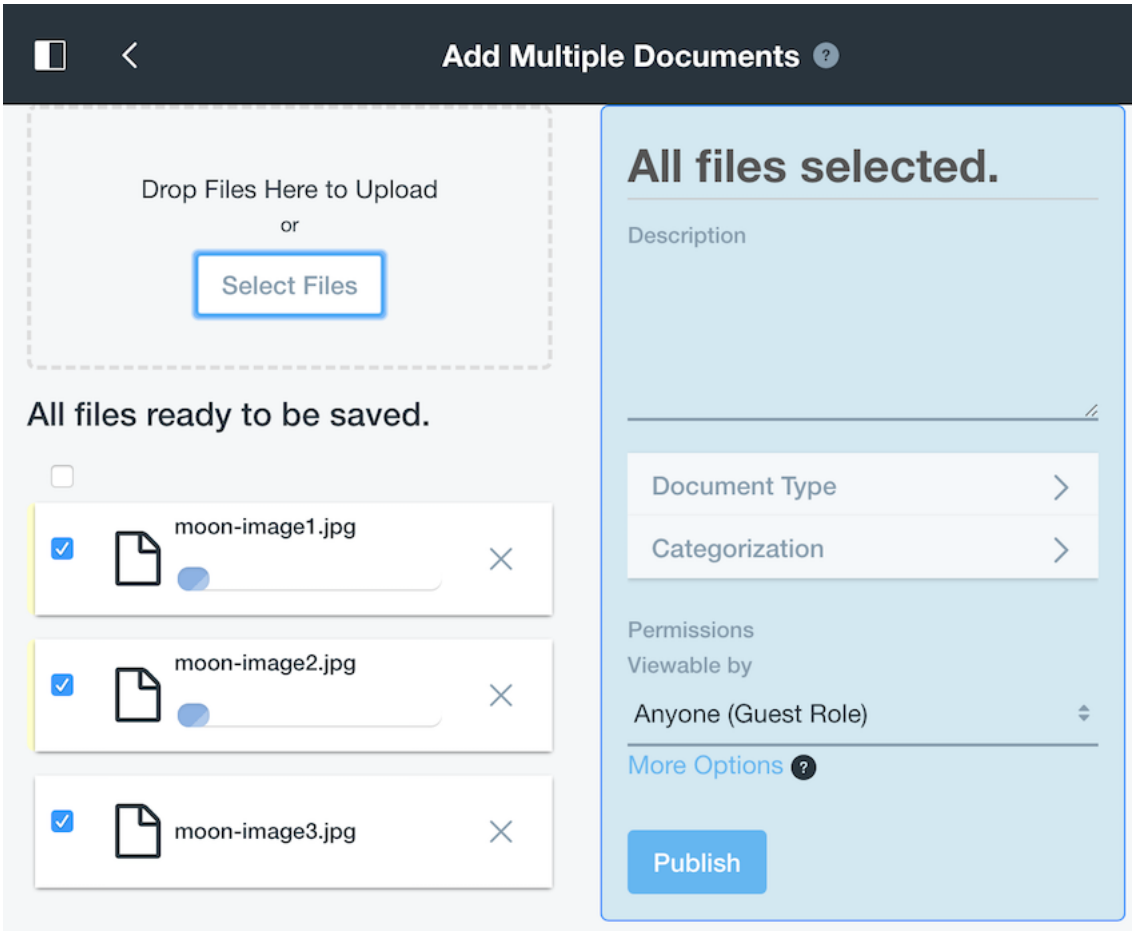


Figure 5.6: The Documents and Media library lets you add multiple documents at once.

4. Click *Save*. This returns you to the Documents and Media library's Home folder, where your new folder is now available.

5. Repeat the steps above to folders named *Frontpage Images* and *Lunar Guides* to the Home folder.

Sweet! The Lunar Resort's Documents and Media library now contains two folders. You'll use them to organize images next.

Organizing Media

Before moving your files into the new folders, it's important to know that you can move files that are currently in use, without issue. For example, if a piece of web content uses an image, and you move that image to a different Documents and Media folder, you don't need to update that web content with the image's new location. Liferay DXP maintains such file connections for you.

Follow these steps to move files to your new folders:

1. Return to the Documents and Media library's Home folder and check the boxes on the images `lunar-resort.png`, `space.png`, and `space-program.png`.

2. Drag and drop the selected images to the Web Content Images folder.

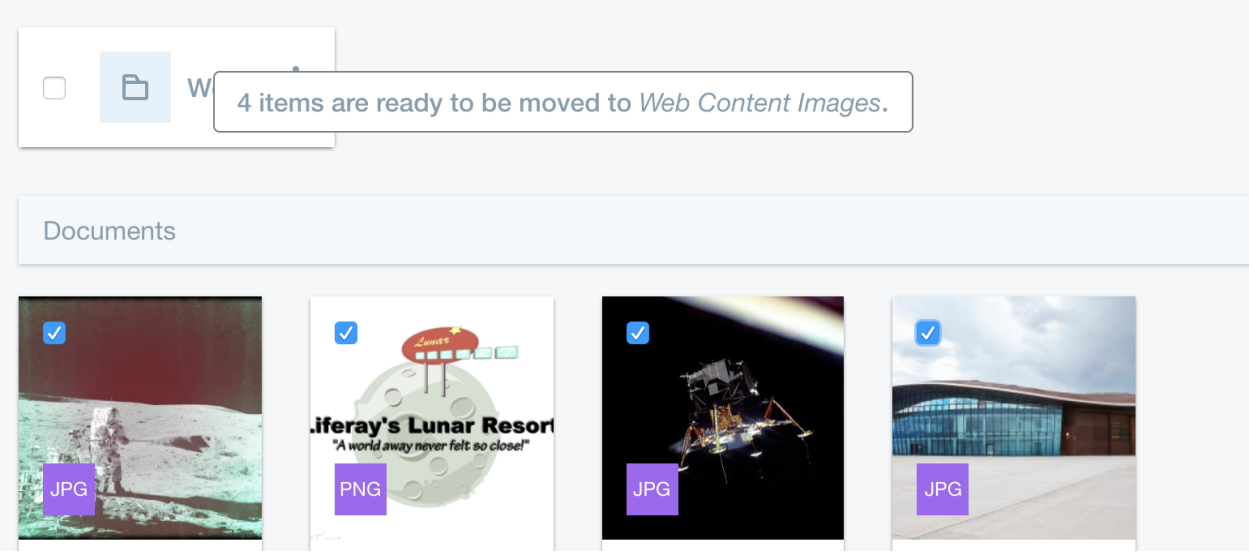


Figure 5.7: The Documents and Media library lets you drag and drop files into a folder.

3. The next screen lets you review and confirm the move. It also lets you change which files are included, and their destination. Leave the default selections and click *Move*.

4. Repeat these steps to move the guide images to the *Lunar Guides* folder and the Home folder's remaining files to the *Frontpage Images* folder.

Well done! Now your Documents and Media library contains the images you need for creating more content. You also learned how to upload and manage files in Documents and Media. Next, you'll learn about creating more complex content and styles using structures and templates.

CHAPTER 6

WEB CONTENT STRUCTURES AND TEMPLATES

Creating Content
Step 3 of 7

In Liferay DXP, you can use *structures* and *templates* to create new web content types and layouts. A structure defines the type of items in your content, such as text, images, calendar items, checkboxes, links, and more. Structures are based on Liferay DXP's forms functionality. When creating content based on a structure, you must fill out that structure's fields.

A template uses a templating language to display a structure's items, so you can apply styles and logic to create complex or interactive content. You can create templates in FreeMarker or Velocity. Templates can contain CSS, HTML, JavaScript, and elements of the templating language it uses.

Before getting started with structures and templates, you should know something important: you've actually already used them! Basic Web Content is a type of web content defined by a structure and a template. Now you'll use structures and templates to create something a little more intricate.

Let's Go!

6.1 Creating Structures

Creating Content
Step 3A of 3D

Now you'll create your first structure. This structure defines Lunar Guides page content: a picture of the guide along with the guide's name and a link to his personal page.

Creating Your First Structure

1. Open the *Menu* ( ) and select *The Lunar Resort* → *Content* → *Web Content*.

2. At the top-right, select *Options* ( ) → *Structures* to see a list of the available structures. Currently, only the default Basic Web Content structure is available.

3. Click the *Add* icon ( ) at the bottom-right of the page to begin creating a new structure.

4. Name the structure *Lunar Guides List*.

Figure 6.1: Content using a template.

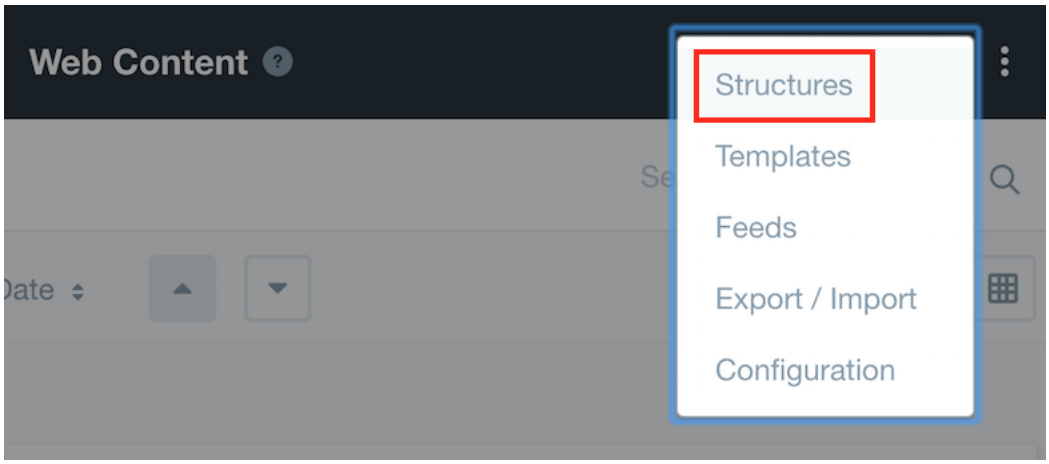


Figure 6.2: Select *Structures* from the *Options* menu.

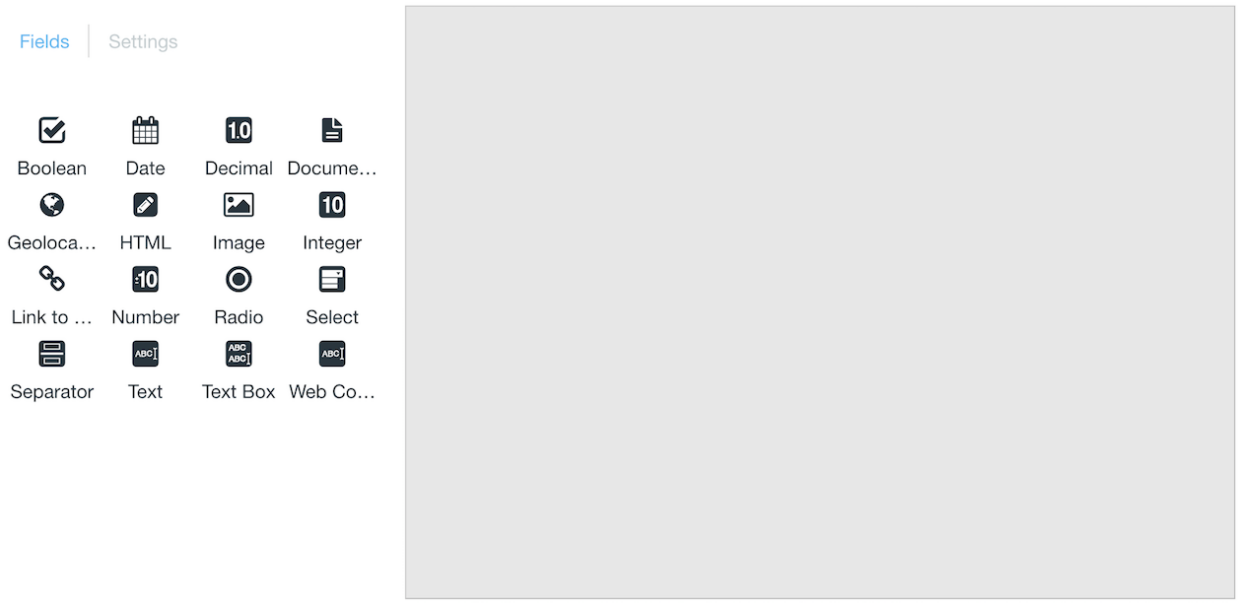


Figure 6.3: These fields are available when creating a structure.

5. Now you're ready to define the structure's fields. You do this by dragging and dropping the available field types from the *Fields* tab to the blank canvas on the right.

   This structure needs a Text field for a title, another Text field for the guide's name, an Image field for the guide's image, and a Link to Page field for the link to the guide's personal page. Follow these instructions to add these fields:

   - Drag a *Text* field onto the canvas.
   - Drag another *Text* field onto the canvas, below the first one.
   - Drag an *Image* field into the second Text field, to nest it under that Text field. This creates a *field group*.
   - Drag a *Link to Page* field into the field group, below the Image field. Be careful to position this Link to Page field so that it's on the same level as the Image field (not nested in the Image field).

6. Click the first Text field. The available fields list is replaced with a list of settings for the selected field. Change the following settings:

   - **Field Label:** This is the field's title that users see when creating content with this structure. Set this to *Title*.
   - **Name:** This is the field's internal name that you can access in a template. Set this to *title*.

7. Click the second Text field. Change its *Field Label* to *Lunar Guide Name*, and its *Name* to `name1`.

8. Click the *Image* field. Change its *Name* to `image1`.

9. Click the *Link to Page* field. Change its *Name* to `link1`.

10. There are four lunar guides, so you need four field groups. Fortunately, you don't have to create each additional field group manually. Hover over the Lunar Guide Name field and click the *Duplicate* button

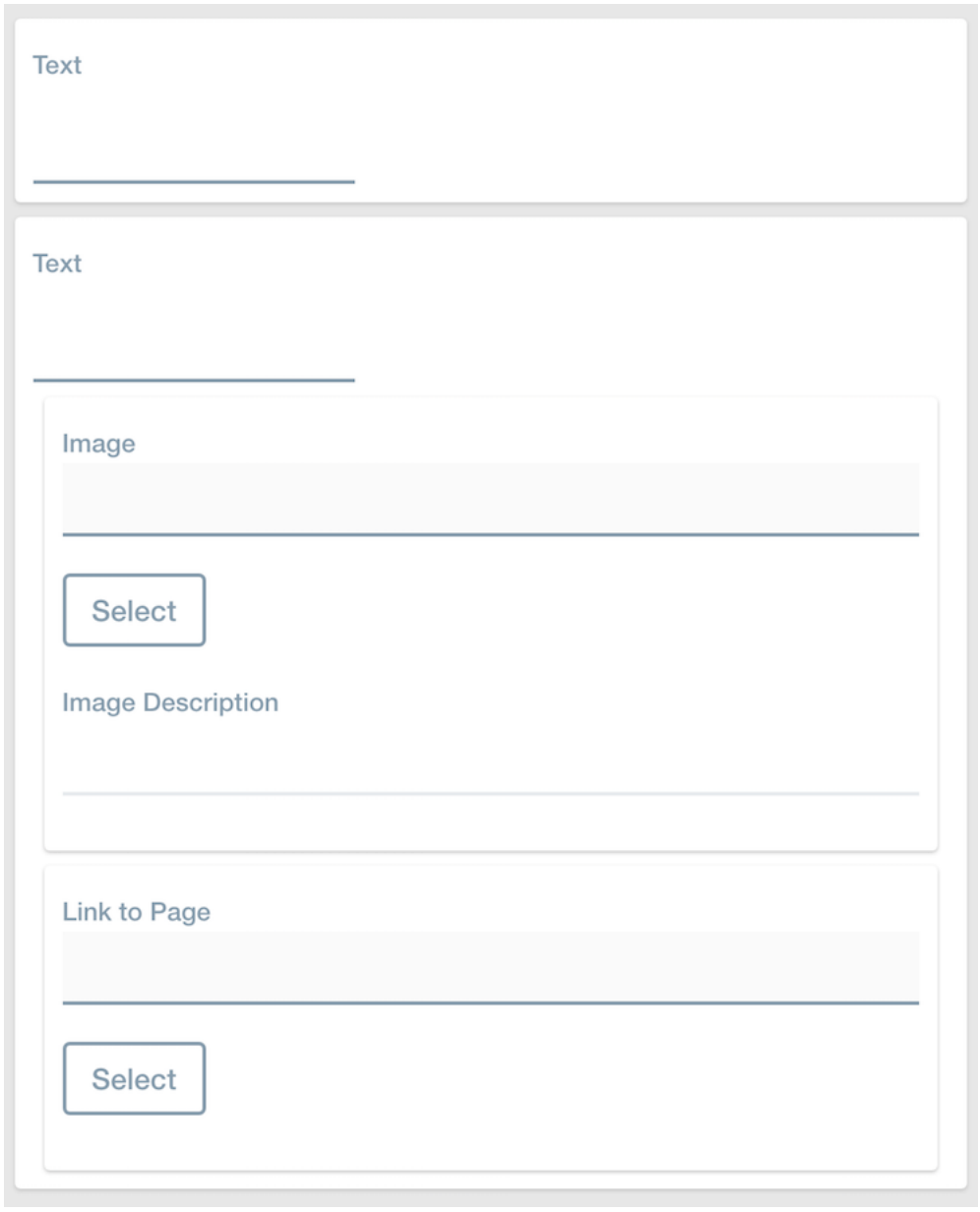


Figure 6.4: The canvas should look like this after you add the Text, Image, and Link to Page fields. Note that the Image and Link to Page fields are nested in the second Text field.

( ) that appears to the field's right. This duplicates the entire field group. Click this button two more times until you have a total of four identical field groups.

```
**Note:** Instead of duplicating fields, you could have used each field's
*Repeatable* option. This lets users decide how many fields or field groups
to use when they create content. This is a more advanced option that you'll
learn about later.
```

---

11. Now you'll change the duplicate fields' names. Repeat steps 7 through 9 above for each duplicate field group, matching the digit in each field name to the field group's number. For example, name the second field group's *Name*, *Image*, and *Link to Page* fields `name2`, `image2`, and `link2`, respectively. Likewise, the digit in each field name should be *3* for the third field group, and *4* for the fourth field group.
12. Click *Save*.

Great! Next, you'll create the template to go along with this structure.

6.2 Creating Templates

```
<p>Creating Content<br>Step 3B of 3D</p>
```

Structures, like the one you created in the previous step, need templates to style and display their items. Now you'll use the FreeMarker templating language to create a template.

Follow these steps to create your template:

1. Open the *Menu* ( ) and select *The Lunar Resort* → *Content* → *Web Content*.
2. At the top-right of the screen, select *Options* ( ) → *Templates*.

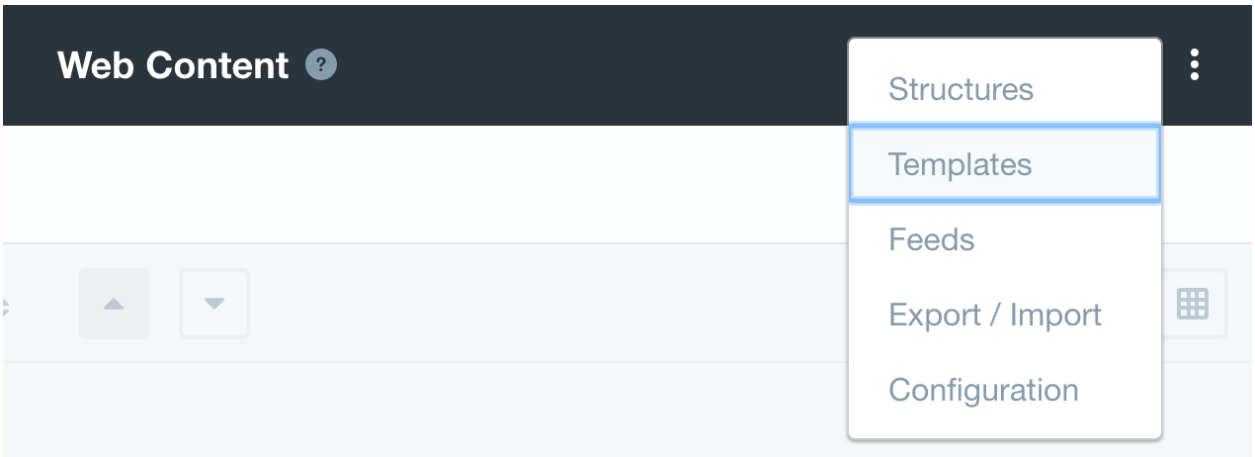


Figure 6.5: Select Templates from the menu.

3. Click the *Add* icon ( ) at the bottom-right of the page to begin creating a new template.

4. Name the template *Lunar Guides List*.

5. Now you must select a structure to link with this template. Open the *Details* section, click the *Select* button under *Structure*, and select the *Lunar Guides List* structure. Click *OK* when prompted.

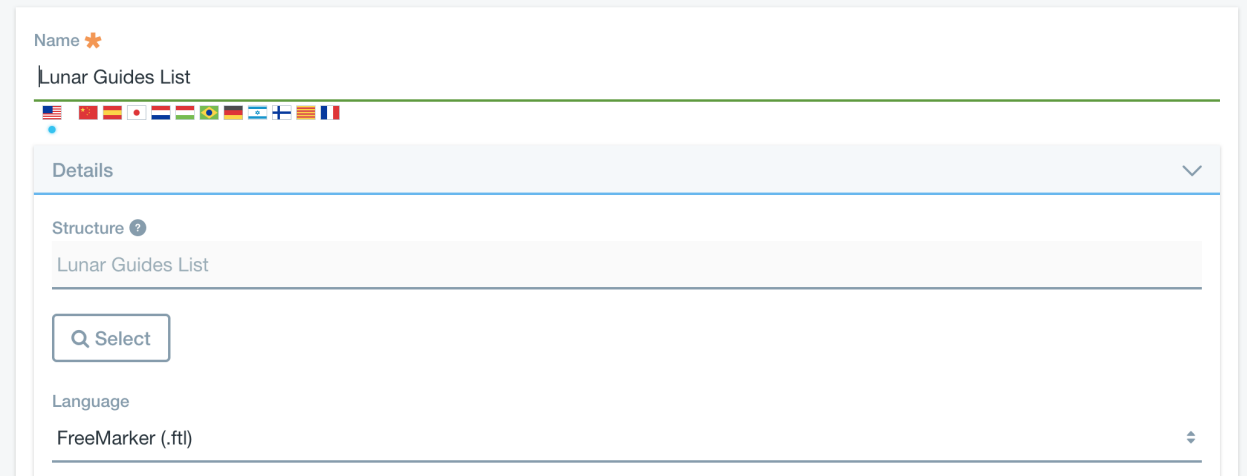


Figure 6.6: Template details.

6. Next, select the template's language. Still in the Details section, make sure *FreeMarker* is selected for the Language field (it should be selected by default). FreeMarker uses the FreeMarker Template Language (FTL), which uses HTML, interpolations (${...}), and FTL tags. FreeMarker templates can also include CSS. For more information on FTL, see the FreeMarker documentation. Note that Liferay DXP also supports Velocity and Extensible Stylesheet Language.

7. Now you're ready to create the script. First you'll insert the CSS. This example uses a four column, two row grid layout to define the field positions. The content appears only in columns two and four—the other two columns exist for spacing. This example also defines the text heading styles. In the *Script* section, replace the code in the editor with this code:

```
<style>
   .wrapper {
  display: grid;
  grid-template-columns: 300px, 300px, 300px, 300px;
  grid-gap: 10px;
  grid-auto-rows: minmax(auto, auto);
  text-align: center;
}

h1 {
    text-align: center;
}

h2 {
   position: relative;
   color: black;
}
```

```
.item-one {
  grid-column: 2;
  grid-row: 1;
  max-width: 355px;
}

.item-two {
  grid-column: 4;
  grid-row: 1;
  max-width: 355px;
}

.item-three {
  grid-column: 2;
  grid-row: 2;
  max-width: 355px;
}

.item-four {
  grid-column: 4;
  grid-row: 2;
  max-width: 355px;
}
</style>
```

8. Now you're ready to create the template's HTML. Like the structure, you can use the sidebar to the editor's left to insert fields in the editor. Doing so automatically inserts fields with the appropriate variables. *General Variables* provides quick access to universal information for your Liferay DXP instance. *Fields* provides access to the variables that your structure defines. Mousing over a field displays its variable name. Clicking a field adds the code in the editor that retrieves that field's data.

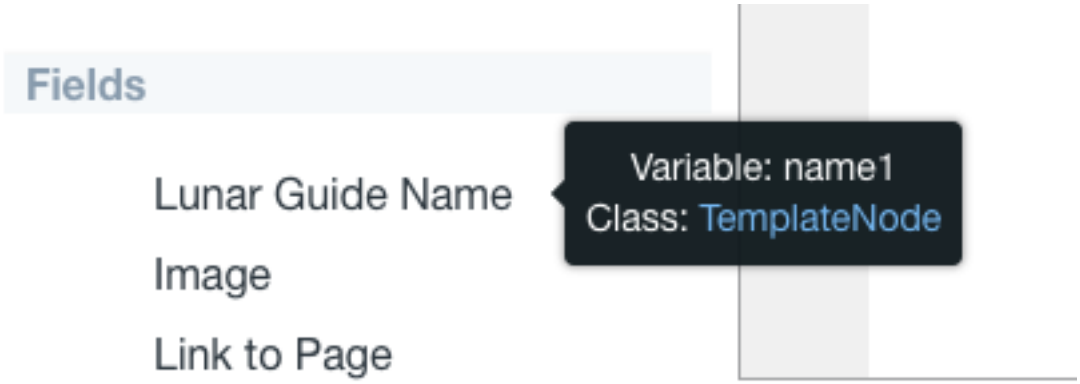


Figure 6.7: A field's tooltip shows that field's variable name.

In this example, however, you'll copy and paste the following code. This code gets the field values that the user entered, styles the text and images, and puts all the information into divs as defined in your CSS. Add this code in the editor, below the closing `</style>` tag:

```
<h1>${title.getData()}</h1>

<div class="wrapper">
  <a class="item-one" href="${name1.link1.getFriendlyUrl()}">
  <h2>${name1.getData()}</h2>
    <#if name1.image1.getData()?? && name1.image1.getData() != "">
      <img alt="${name1.image1.getAttribute("alt")}" src="${name1.image1.getData()}" />
    </#if>
  </a>
```

```
  <a class="item-two" href="${name2.link2.getFriendlyUrl()}">
  <h2>${name2.getData()}</h2>
    <#if name2.image2.getData()?? && name2.image2.getData() != "">
      <img alt="${name2.image2.getAttribute("alt")}" src="${name2.image2.getData()}" />
    </#if>
  </a>

<hr />

  <a class="item-three" href="${name3.link3.getFriendlyUrl()}">
  <h2>${name3.getData()}</h2>
    <#if name3.image3.getData()?? && name3.image3.getData() != "">
      <img alt="${name3.image3.getAttribute("alt")}" src="${name3.image3.getData()}" />
    </#if>
  </a>

  <a class="item-four" href="${name4.link4.getFriendlyUrl()}">
  <h2>${name4.getData()}</h2>
    <#if name4.image4.getData()?? && name4.image4.getData() != "">
      <img alt="${name4.image4.getAttribute("alt")}" src="${name4.image4.getData()}" />
    </#if>
  </a>

</div>
```

9. Click *Save*.

Great! Now you have a template for displaying your structure's items. Next, you'll create some content that leverages your structure and template.

6.3 Creating Content with Structures and Templates

```
<p>Creating Content<br>Step 3C of 3D</p>
```

Now that you have a template for your structure, you can use that structure to create content. Remember, the template formats the structure's content for display.

Follow these steps to create your content:

1. Open the *Menu* () and select *The Lunar Resort* → *Content* → *Web Content*.

2. Click the *Add* icon () at the bottom-right of the page and select *Lunar Guides List*. This takes you to the New Web Content form for your structure. Note that the fields and field groups you defined in the structure are all here.

3. For both *Title* fields, enter *Choose Your Lunar Guide*. Don't worry about the duplication. The first Title field is default for all web content. You'll use the Web Content Display app to show only your structure's title field. This lets you use the template to style the field's content, rather than apps that display web content.

4. Fill out a field group for each lunar guide: *Cody*, *Jim*, *Russ*, and *Steve*. For each *Image* field, select the guide's image from the images you uploaded earlier. For each *Link to Page* field, click *Select* to open the dialog that lets you select a page in your site. In this dialog, click *Lunar Guides*, select the guide's page, and click *Select*.

5. Click *Publish*.

The content now exists, but you must still publish it on a page in your site. Follow these instructions to do so:

1. Go to the *Lunar Guides* page.
2. Click the *Add* icon ( ) at the top-right of the page and expand *Content*.
3. Drag the content *Choose Your Lunar Guide* to the page. This displays your content in a Web Content Display app.
4. Note that your content contains duplicate titles. You'll fix this now. Click the Web Content Display app's *Options* button ( ) and select *Look and Feel Configuration*. In the *General* tab, select *Barebone* for *Application Decorators*, then click *Save*. The only title that remains is the one you defined in your structure and styled in your template.
5. Now you can test the content to see how it works. Click a guide's picture or name to go to that guide's page in the hierarchy.

Great! Next, you'll learn how to integrate JavaScript into your templates.

6.4 Advanced Templates

```
<p>Creating Content<br>Step 3D of 3D</p>
```

The template you created for lunar guides used CSS, HTML, and FreeMarker to style and format the corresponding structure's fields. Templates in Liferay DXP can also use JavaScript. This lets you create complex, interactive content. To illustrate this, you'll create a new structure, and then create its template using JavaScript.

Creating the Structure

The structure for the booking form is relatively simple–it only needs a Text field and an Image field. Follow these steps to create it:

1. Open the *Menu* ( ) and select *The Lunar Resort* → *Content* → *Web Content*.
2. At the top-right of the screen, select *Options* ( ) → *Structures*.
3. Click the *Add* icon ( ) at the bottom-right of the page.
4. Name the structure *Booking Form*.
5. Add a *Text* field to the canvas. Below that field, add an *Image* field.
6. Set the Text field's *Field Label* to *Button Text*. Set the field's *Name* to `buttontext`.
7. Set the Image field's *Name* to `bgimage`. Leave the field's Field Label alone.
8. Click *Save*.

Now that the structure is complete, you can move on the template.

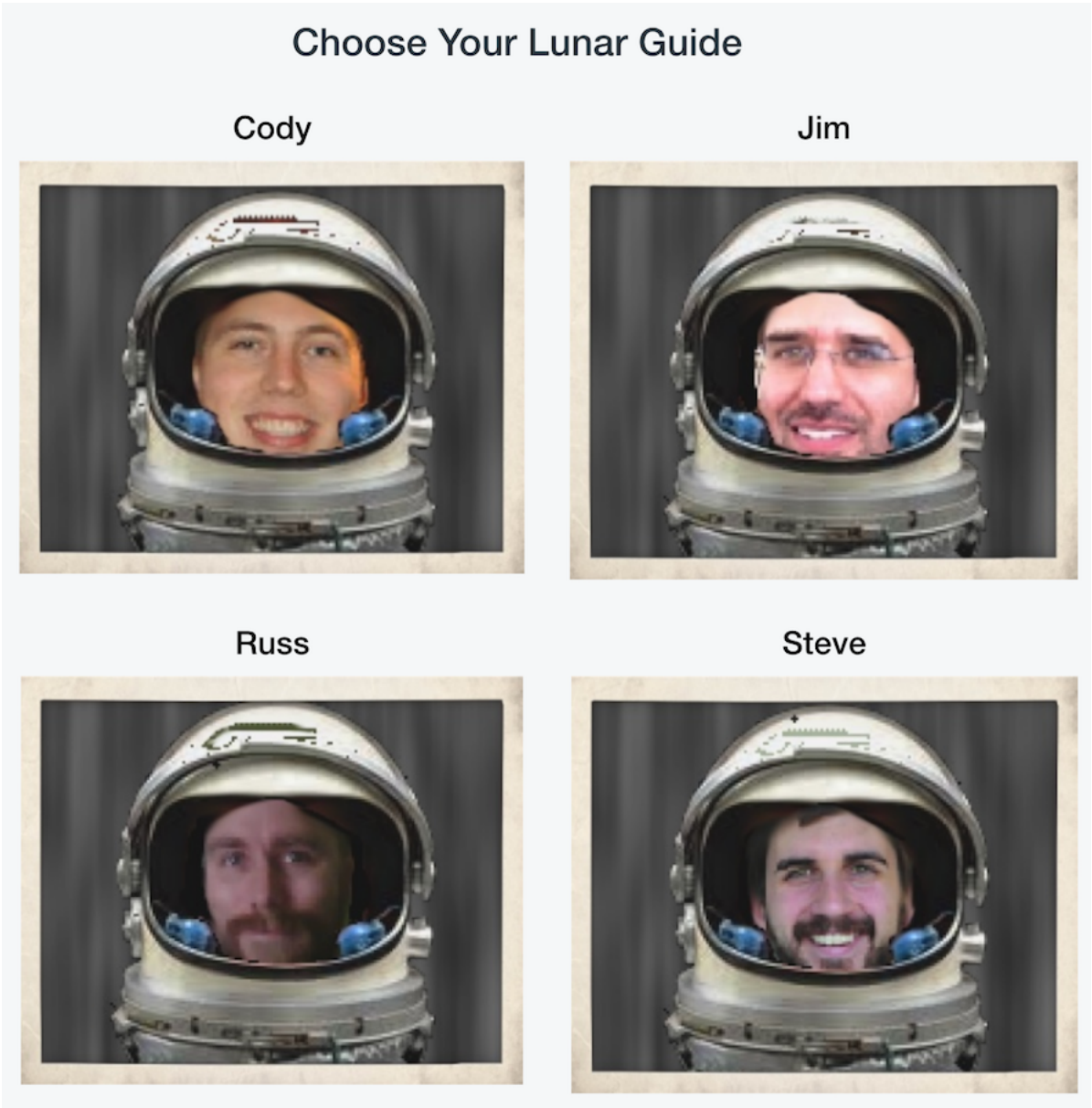


Figure 6.8: The lunar guides, at your service!

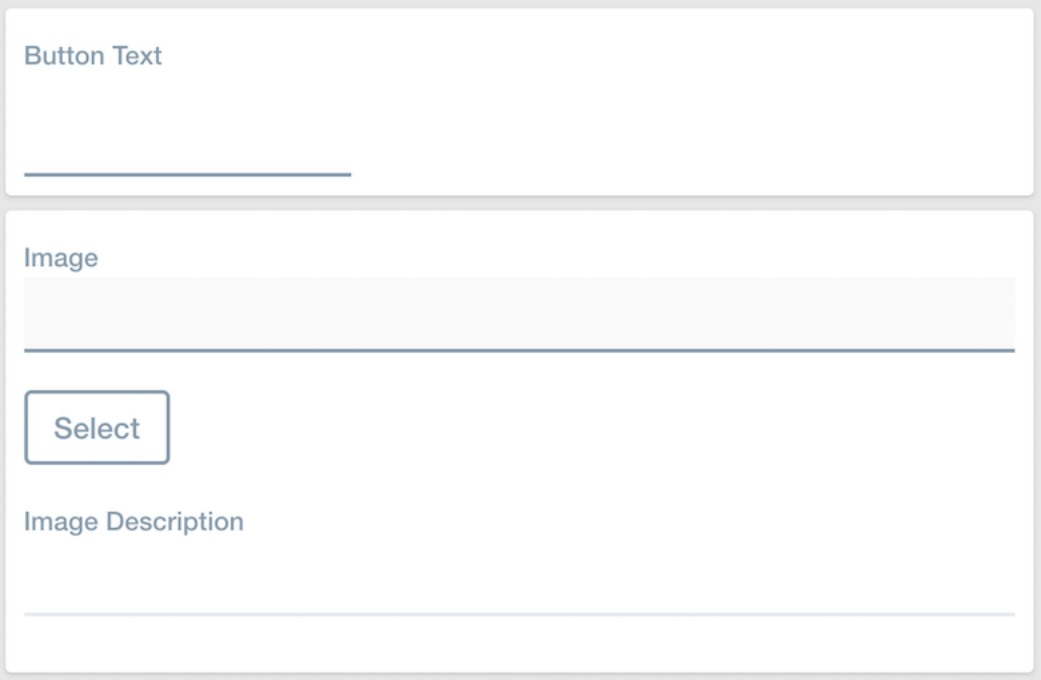


Figure 6.9: The Booking Form structure contains a Text field above an Image field.

Creating the Template

The booking form's template styles the text and image, and also contains a JavaScript function that submits the booking. Follow these steps to create this template:

1. Open the the *Menu* ( ) and select *The Lunar Resort* → *Content* → *Web Content*.
2. At the top-right of the screen, select *Options* ( ) → *Templates*.
3. Click the *Add* icon ( ) at the bottom-right of the page.
4. Name the template *Booking Form*.
5. Open the *Details* section and select *Booking Form* for the *Structure*. In the *Language* field, leave *FreeMarker* selected.
6. Now you're ready to create the template's script. First, you'll add a JavaScript function that triggers a pop-up when called. In the *Script* section, delete any existing code and add this code:

```
<script>
function bookNow() {
  var popup = document.getElementById("myPopup");
  popup.classList.toggle("show");
}
</script>
```

7. Now you'll add a `div` named `popup` containing a `span` named `popuptext` that operates on the `bookNow` function when clicked. This `span` retrieves the `buttontext` from the structure and sets the `popuptext` to display in the function. Everything is namespaced so that you can create the styles accordingly. Add this code now, **above** the `bookNow` function:

```
<div class="popup" onclick="bookNow()">
  <button id="bookBtn">${buttontext.getData()}</button>
  <span class="popuptext" id="myPopup">You have successfully booked your trip!    <br />  See you on the moon!</span>
</div>
```

8. Finally, you'll add the styles. The following code first defines the styles for the main `div` and the button. Next, it defines the style for the area that appears when the `bookNow` function is called (note that its initial visibility is `hidden`). Finally, when the function runs, the `.show` class becomes active and the visibility is set to `visible`. Add this code **above** the existing code in the editor:

```
<style>
.popup {
    text-align: center;
    width: 100%;
    margin: auto;
    display: inline-block;
    cursor: pointer;
    height: 350px;
    background-image: url("${bgimage.getData()}");
}

.popup button#bookBtn {
    position: relative;
    top: 110px;
    padding: 20px 20px;
    text-align: center;
    text-decoration: none;
    border: none;
    font-size: 40px;
    border-radius: 6px;
    background-color: #65b6f0;
    color: white;
}

.popup .popuptext {
    visibility: hidden;
    width: 450px;
    background-color: #555;
    color: #fff;
    text-align: center;
    border-radius: 6px;
    padding: 8px 0;
    position: absolute;
    z-index: 1;
    bottom: 100%;
    left: 50%;
    margin-left: -220px;
    font-size: 150%;
}

.popup .show {
    visibility: visible;
}
</style>
```

When you're done, click *Save.*

Awesome! Now you have a template for the booking form. You're ready to use the structure and template to create some content.

Creating the Content

Like the structure, content for this structure and template is simple. It can only contain an image and button text. Follow these steps to create such content now:

1. Open the the *Menu* ( ) and select *The Lunar Resort* → *Content* → *Web Content*.
2. Click the *Add* icon ( ) at the bottom-right of the page and select *Booking Form*.
3. Enter *Booking Form* for the title.
4. Enter *Book Now!* for the button text.
5. For the image, click *Select* and then upload and select the `booking-image.png`.
6. Click *Publish*.

Now, all you need to do is add it to the page.

Displaying the Content

1. Navigate to the Lunar Resort's *Book a Trip* page.
2. Click the *Add* icon ( ) at the top-right of the page and expand *Content*.
3. Drag the *Booking Form* content onto the page.
4. Open the *Look and Feel Configuration* of the Web Content Display app that contains your content.
5. On the *General* tab, set *Application Decorators* to *Barebone* and click *Save*.
6. Refresh the page, and then test out your button.

Figure 6.10: Template before click.

Great! Now you know how to use structures and templates to create and style web content. Next, you'll learn about some other ways to publish content in Liferay DXP.

Figure 6.11: Template after click.

6.5 Publishing Content with Content Display Pages

```
<p>Creating Content<br>Step 4 of 7</p>
```

So far, you've manually published most of your site's content. Now you'll use content display pages to publish content automatically. Liferay DXP's Asset Publisher app makes this possible. You can configure the Asset Publisher to automatically publish any type of asset (e.g., images, blogs, web content). A content display page is simply any page that contains an Asset Publisher app configured to display the page's associated content. Because the Asset Publisher is highly configurable, you can create a content display page that displays one or many pieces of content.

First, you'll create a simple content display page for displaying a single piece of content.

Creating a Content Display Page

Follow these instructions to create the content display page:

1. In the *Menu* ( ), open *The Lunar Resort* → *Navigation*.
2. Click the *Options* button ( ) next to *Public Pages* and select *Add Public Page*.
3. Name the page *Hazard Disclaimer*.
4. Set the *Hide from Navigation Menu* toggle to *YES*. This hides the page from the main navigation, meaning that users can only access the page directly via a link. This means that to the user, the page is synonymous with its content.
5. In the *Type* selector, select *Templates* → *Content Display Page*.
6. Set the *Inherit Changes* toggle to *NO*.
7. Click *Add Page*.

Nice! Now you're ready to create the page's content.

Creating the Display Page's Content

Use these steps to create the page's content:

1. In the *Menu* (), select *The Lunar Resort* → *Content* → *Web Content*.
2. Click the *Add* icon () at the bottom-right of the page and select *Basic Web Content*.
3. Name the content *Hazard Disclaimer*.
4. In the *Content* field, enter the following:

```
Potential Hazards of Space Travel and Lunar Exploration include but are not limited to:

Accidental ejection from airlock into the cold vacuum of space
Spacesuit failure
Spacecraft engine explosion
Running and jumping so fast that you achieve escape velocity and are no longer bound by the moon's gravity
Accidental exposure alien spores or eggs
Collisions with stray meteorites
Excess exposure to solar radiation
Muscular atrophy

Neither the The Space Program, The Lunar Resort nor any of the their subsidiaries are responsible for injury or harm caused by these or similar acci
```

5. Open the *Display Page* section and click *Choose*.
6. Select the *Hazard Disclaimer* page, and click *Done*.
7. Click *Publish*.

Awesome! Now that your content exists on the display page, you'll configure a link to it so that users can view it.

Using Content Display Pages

You'll display the content via a link in an Asset Publisher. To do so, follow these steps:

1. Go to the *Book a Trip* page and use the *Add* menu () to add an Asset Publisher app to the page.
2. Click the Asset Publisher's title (just beneath the portlet bar that appears on mouse-over) to edit it and change it to *Waivers and Disclaimers*.
3. Open the Asset Publisher's *Configuration* dialog.
4. Under *Asset Selection* choose *Manual*.
5. Open the *Asset Entries* section and click *Select* → *Basic Web Content*.
6. Select *Hazard Disclaimer*.
7. Click *Save*.
8. Still in the Asset Publisher's *Configuration* dialog, click the *Display Settings* tab.
9. In *Asset Link Behavior*, select *View in Context*.

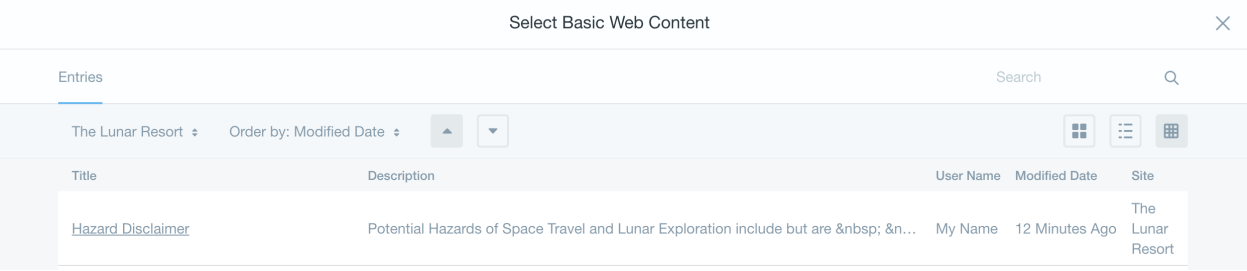


Figure 6.12: Selecting individual content for display.

10. Click *Save* and then close the configuration dialog.

A summary of your content now appears in the Asset Publisher, with a link that takes you to the content display page containing your full content. The content display page also provides a friendly URL to your content that you can share.

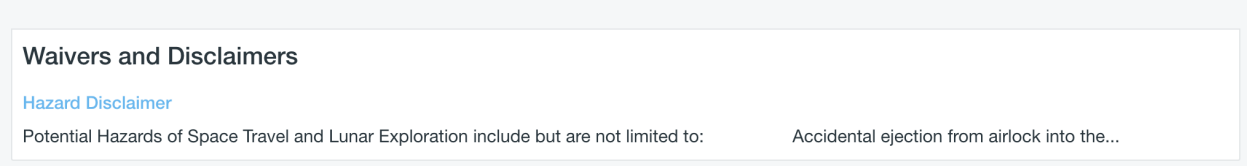


Figure 6.13: Your content now appears in the asset publisher.

Right now, you're using the content display page's default configuration. You can also configure the page to look however you want, or turn an existing page into a content display page. That's what you'll do next.

6.6 Advanced Content Display Page Options

```
<p>Creating Content<br>Step 5 of 7</p>
```

Now that you've used a basic use case for the content display page, let's take a deeper look at how they work and how to use them for more advanced use cases.

Choose Your Own Content Display Page

A Content Display Page can be any page with an Asset Publisher configured as the "default" asset publisher for a page, and you can set up the rest of the page however you like. This allows you to use Content Display Pages for things beyond displaying a single static piece of content or providing a friendly URL.

Configuring Asset Publishers Now, let's configure an Asset Publisher to function as our primary content display.

1. Go to the Activities page.

2. Open the Configuration menu for the Asset Publisher on the right side.

3. In *Asset Selection* scroll down to *Filter* and open it.
4. Set *Show only assets with Activities as its display page* to *Yes*.
5. Open *Display Settings*
6. For the *Display Template* select *Full Content*.
7. For the *Number of Items to Display* change it to 1.
8. Set *Set as the Default Asset Publisher for This Page* select *Yes*.

   This is the most important step in configuring a Content Display Page. If a page has an Asset Display Template set as the default for the page, any content can be displayed in it.
9. Set *Show Metadata Descriptions* to *No*.
10. Click Save.

Most of the configuration options are self explanatory. Next, configure the Asset Publisher on the left to function as an Asset Selector.

1. Open the Configuration menu.
2. In *Asset Selection* scroll down to *Filter* and open it.
3. Set *Show only assets with Activities as its display page* to *Yes*.
4. Open *Display Settings*
5. For *Display Template* select *Title List*.
6. For *Asset Link Behavior* select *View in Context*
7. Click Save.

The Asset Publishers are configured, but there's no content configured for this page. Next you'll create content that uses a Content Display Page.

Creating Content with a Display Page

Any content can have a display page set for it, and any page can be a display page if it has a properly configured Asset Publisher on it. Next, you'll create three content items to be displayed here.

1. Go to the *Web Content* page in *Site Administration*.
2. Click the + and select *Basic Web Content*.

You'll need to repeat this three times. The content will only have a title and an image.

1. For the first item, set the *Title* as "Lunar Rover Racing".
2. Add the `lunar-rover.png` image.
3. Scroll down and for Display page, click *Choose* and select *Activities*.

4. Click *Publish*.
5. For the second item, name it "Lunar Golf" and add the `lunar-golf.png` image, set the Content Display Page to *Activities* and click *Publish*.
6. For the third item, name it "Lunar Spelunking" and add the `lunar-spelunking.png` image, set the Content Display Page to *Activities* and click *Publish*.

Housekeeping and Final Test

Now to add a few finishing touches.

1. Go back to the *Activities* page.
2. In the lefthand Asset Publisher, click the Portlet Title, and change it to *Choose an Activity* and click the checkmark.
3. On the right side, open the *Look and Feel Configuration* and change the *Decoration* to *Barebone*.

Now that your page looks great, test it out -

1. Click on a title and that content will display in the publisher to the right.

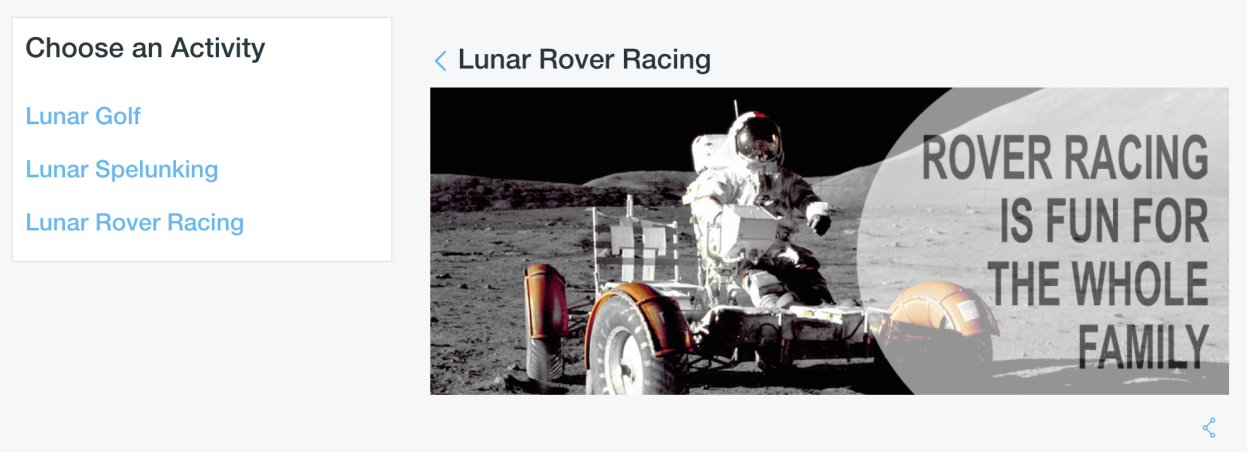


Figure 6.14: Final Activities Page.

Congratulations, you've finished creating the Activities page, and configured a dynamic content display.

6.7 Application Display Templates

```
<p>Creating Content<br>Step 6 of 7</p>
```

For the final piece of the site that you'll construct with Liferay's Web Experience Management system, you need to create a more engaging front page graphic. To do this, you'll use multiple images and an Application Display Template (ADT) to display them as a carousel.

ADTs are essentially the same as Web Content Template, but rather than have the variables and fields defined by a structure, they're defined by the application that you're applying the template to.

ADTs and Scope

Liferay comes bundled with several default ADTs which exist at the *Global* scope. Scope, in Liferay DXP simply means where content can be used or viewed. All of the content that you have created so far has been in the scope of The Lunar Resort site. If you were to create a second site on the same Liferay server, you would not be able to access content that you created for The Lunar Resort in the second site, but everything created in the Global scope is available for both.

To create your own ADT, switch to the Global scope and create it there alongside all of the system default ADTs.

Creating the ADT

1. Open the main menu.
2. Next the *The Lunar Resort*, click the compass icon to open the *Select Site* dialog.
3. From there, click *My Sites* and then select the *Global* site.

Everything you create in this context will be for the Global scope. Now to create the ADT.

4. Click on *Configuration → Application Display Template*.
5. Click the + button and select *Documents and Media Template* to add yours to the list.
6. Name the template *Frontpage Carousel*.
7. In the code area, paste in the styles first:

```
<style>
      .slides {
        margin: auto;
        width: 100%;
        height: auto;
        border-radius: 5%;
      }

      h1 {
       position: absolute;
       color: white;
       top: 100px;
       width: 100%;
       text-align: center;
       font-size: 50px;
      }
</style>
```

You have a style which manages the slides and a style that manages the heading text.

8. Then add the main HTML section:

```
<#if entries?has_content>
    <div id="<@portlet.namespace />carousel">
        <#assign imageMimeTypes = propsUtil.getArray("dl.file.entry.preview.image.mime.types") />

        <#list entries as entry>
            <#if imageMimeTypes?seq_contains(entry.getMimeType())>
              <img class="slides" src="${dlUtil.getPreviewURL(entry, entry.getFileVersion(), themeDisplay, "")}">
            </#if>
```

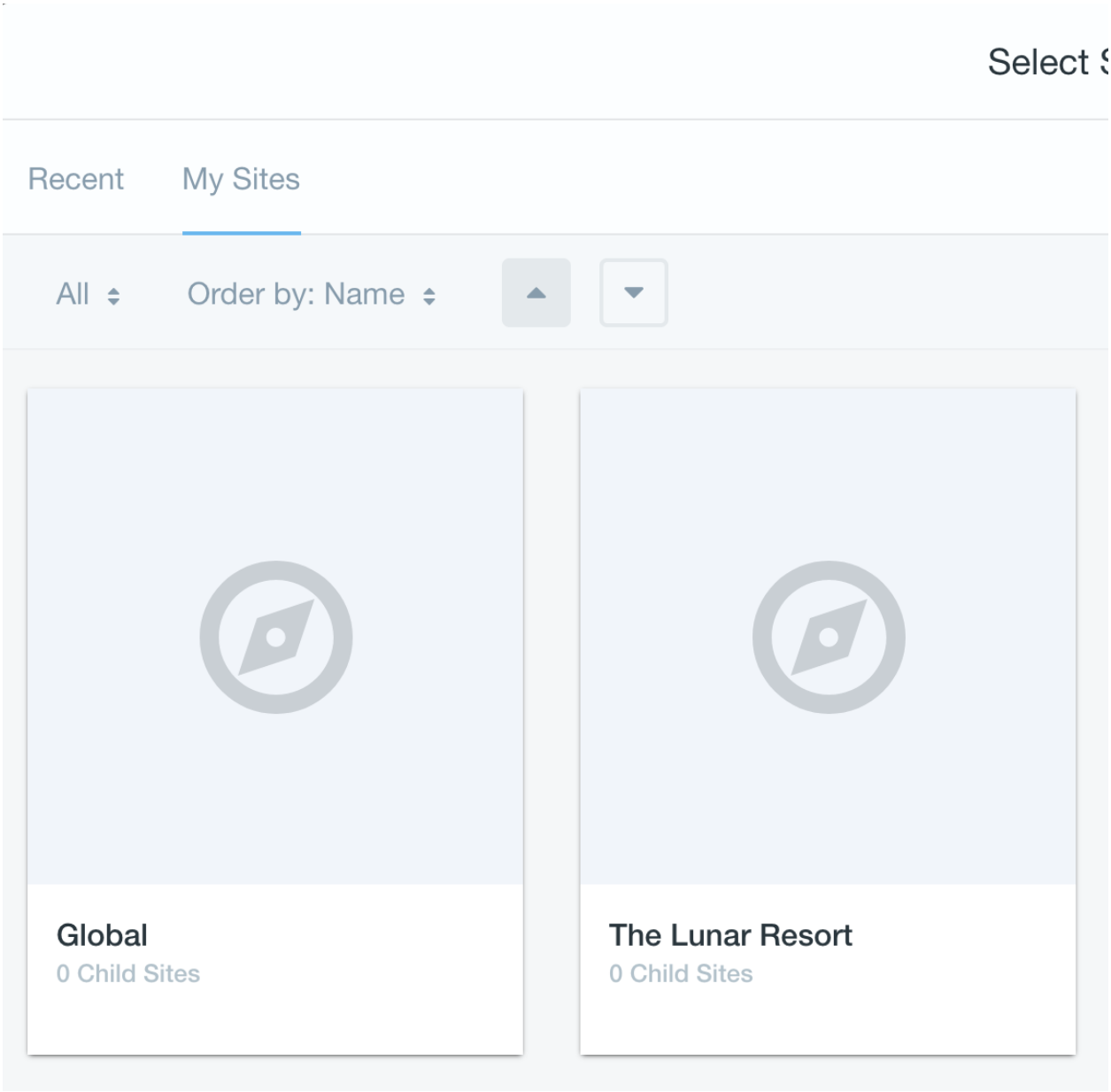


Figure 6.15: Site selection.

```
        </#list>
            <div class="flex-container" style="height: 100%">
                <h1>Make Memories at the Lunar Resort<h1>
              <div class="flex-item-center text-center" style="width: 100%" />
            </div>
    </div>
```

This calls the script (which you'll add momentarily) that sets the mime type for display (images), gets the list of `entries` that will be displayed, and creates the container. It's all wrapped in an `#if` statement which will only display this block if there are entries to display.

9. Next add the script:

```
<script>
    var slideIndex = 0;
        carousel();
    function carousel() {
    var i;
    var x = document.getElementsByClassName("slides");
    for (i = 0; i < x.length; i++) {
      x[i].style.display = "none";
    }
    slideIndex++;
    if (slideIndex > x.length) {slideIndex = 1}
    x[slideIndex-1].style.display = "block";
    setTimeout(carousel, 3000); // Change image every 3 seconds
    }
</script>
</#if>
```

This is a fairly straightforward Javascript carousel. You get the slides, and iterate through them to display them.

Once you Save the template, you're all done creating the ADT. Now let's go set it up.

Using the ADT

To use the ADT, you'll add a Media Gallery to the page, point it at the appropriate folder, and set it to use the display template we created.

1. Go back to the Lunar Resort homepage.
2. Remove the Web Content Display from earlier by clicking on the portlet menu and selecting *Remove*, then *OK* when prompted.
3. Open the add menu, go to *Applications* and add a *Content Management → Media Gallery* to the page.
4. Open the portlet menu for the Media Gallery and select *Configuration*.
5. Scroll down to *Folders Listing* and click *Select*.
6. Click *Choose* to select the *Frontpage Images* folder.
7. For *Display Template* select *Frontpage Carousel*
8. Click *Save*.
9. Now go to *Look and Feel Configuration* and set the *Application Decorator* to *Barebone*

Good work! Your Welcome page is complete!

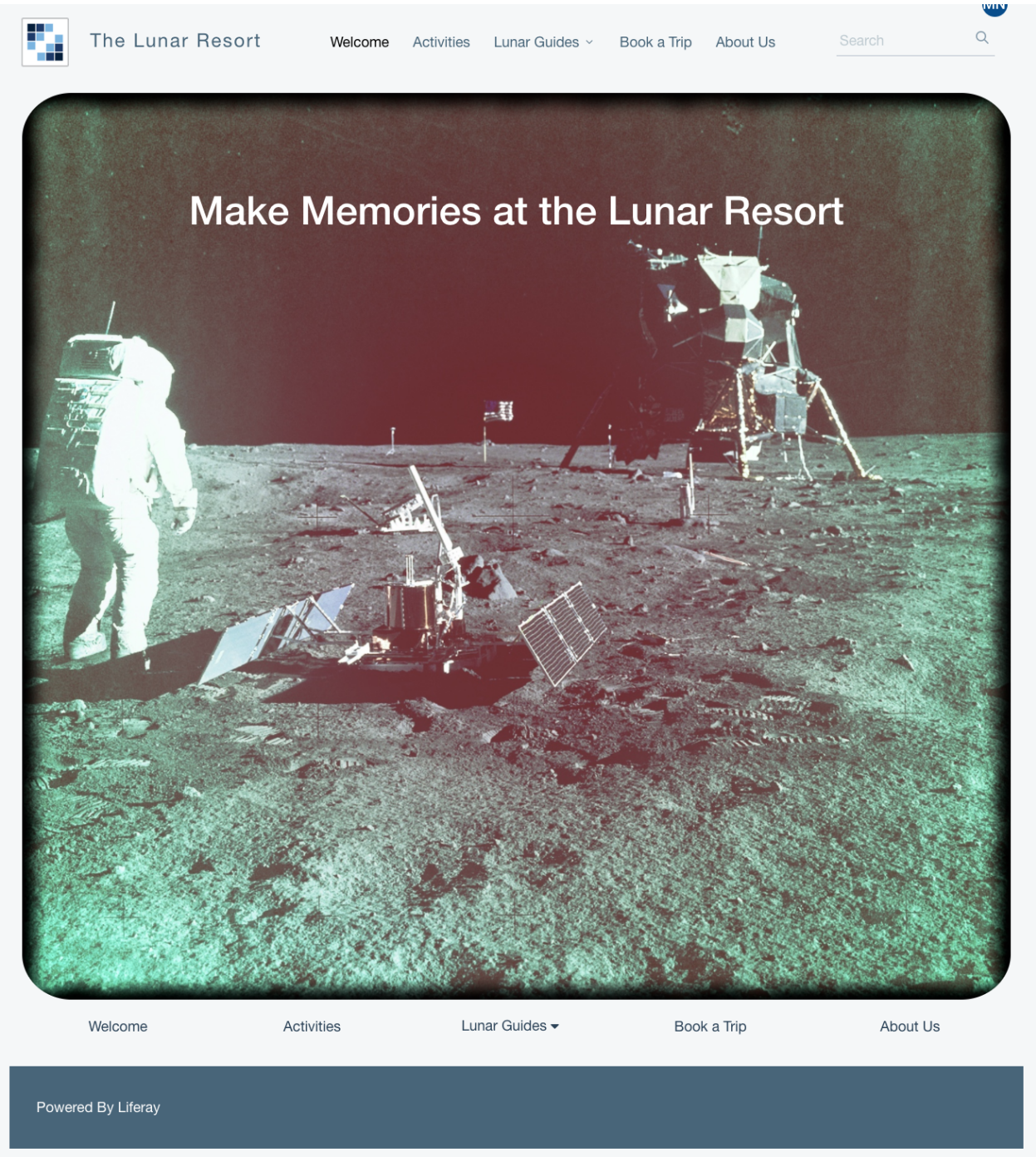


Figure 6.16: Site selection.

6.8 Review Your Site

Creating Content
Step 7 of 7

You create the pages, you created the content, and you organized and displayed it all using Liferay's Web Experience Management tools. It's time to take a step back and look at your site.

Now that you have the main content completed, your next step will be to use Liferay's Collaboration tools, like Blogs, the Calendar, and Message Boards to connect more powerfully with your users.

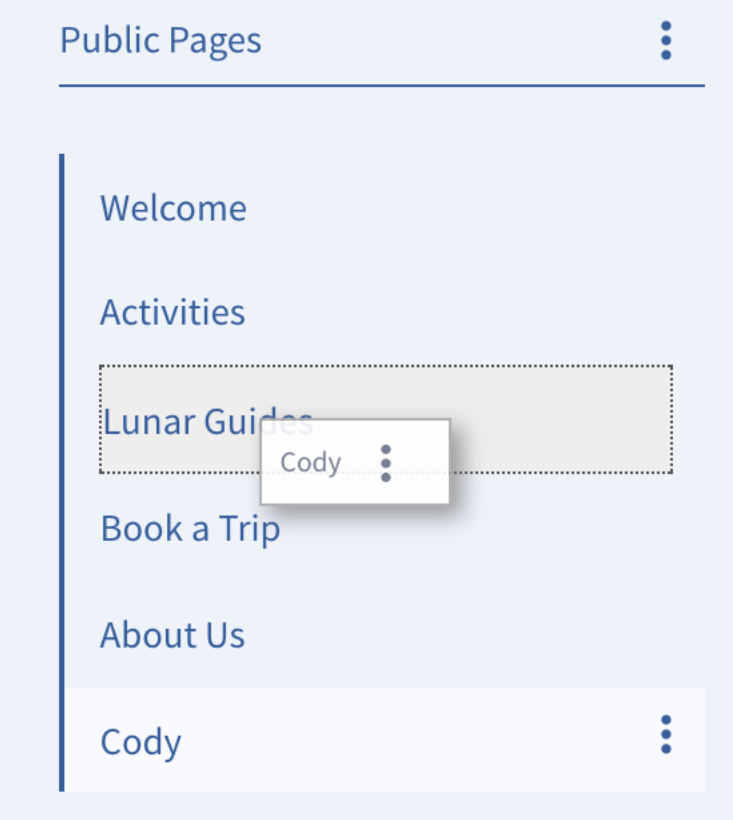


Figure 6.17: IMAGE

Chapter 7

Web Experience Management

Experience; consider that word for a moment. Not the type of experience you gain with repetition, but the contact or encounter you have with something. For example, if you're hiking in the woods and are mauled by a black panther, you would consider this a *bad* hiking experience. Suppose you travel to your local food market and someone gives you a large amount of money for no reason other than to be nice. You would consider this a *great* grocery shopping experience.

These two examples may be a bit extreme, so let's tone this down to something a little more practical. Suppose you're a huge sports fan and you're contemplating whether to buy tickets to the big game next week. What are some of the things you might consider that would determine whether you attend the game or not? The *price* of the tickets is usually the most important factor. Others may include parking accessibility, distance to the game, quality of your favorite team, and other factors. To determine whether you attend the game, you weigh the negative factors, like price and the stadium's nightmarish parking setup, with the reason you're attending the game, the *experience*. If you expect the experience of seeing your favorite team play to trump the negative factors, you'll attend. If you expect the experience will be bad or worse than the negative factors you expect to encounter, you'll stay home and watch the game on TV.

Liferay DXP takes the *experience* factor very seriously when it comes to web management. Just like the sports fan that weighs the experience vs. the live game headaches, the user/administrator also weighs the experience of using a web platform with the resulting site created on it. With the apps and features provided with Liferay DXP's Web Experience Management suite, the management of sites, pages, and content is seamless.

Managing sites and pages is a fundamental pillar of any web site, and Liferay DXP provides easy-to-use features that make site/page creation quick and easy. Web Content is another building block upon which most sites rely. With Liferay's web content apps, you can easily create simple or complex content for your site's visitors. Once you've created content, Liferay DXP also provides several innovative ways to publish that content, so users can find the right information as fast as possible. Sometimes your site must adapt on the fly, whether it be for different holidays or events. A testing environment where you can plan the look and feel of your pages/content gives you peace of mind in knowing that you won't have to trial and error site changes on a live page.

Just as the die-hard fan knew he'd have a great experience when buying his tickets to the big game, you can expect a great and seamless web experience with Liferay DXP.

Chapter 8

Starting Site Development

A site is a set of pages that can be used to publish content or applications. Sites can be independent or they can be associated with an organization and serve as the website for that organization. With Liferay DXP, you can create as many different sites as you like within the context of a single Liferay instance.

You can use sites in Liferay DXP to build many different kinds of websites. Whether you're building a large corporate website, a company intranet, or a small site designed to facilitate collaboration among team members, Liferay's framework provides all the tools you need.

You'll begin your journey with a tour of Liferay DXP's user interface. Next, you'll create a custom Lunar Resort Example instance and explore ways to create sites and pages for that Liferay instance. There are many different ways to manage your sites and pages, which is also covered. To begin site development with Liferay DXP, continue on to the next section.

8.1 Creating Sites

A site contains a set of pages that can be used to publish content or applications. By default, Liferay DXP starts with a single site that has a single page. You can build any website you wish out of this, complete with multi-nested page hierarchies. The site can stand independently or can be associated with an organization to serve as the website for that organization. Liferay's framework provides all the site-building tools you need to manage a successful site.

Understanding Site Management

Whether you're building a large corporate website or a small site designed to facilitate collaboration among team members, supporting different kinds of collaboration and social scenarios is a must. Liferay's sites provide three membership types:

Open: Users can become members of the site at any time.

Restricted: Users can request site membership but site administrators must approve requests in order for users to become members.

Private: Users are not allowed to join the site or request site membership. Site administrators can still manually select users and assign them as site members.

In addition to these memberships, when a site is associated with an organization, all the users of that organization are automatically considered members of the site.

You can view all the available open and restricted sites by adding the My Sites application to a page and accessing the *Available Sites* tab. You can request access to any of the sites you're not already a member of by selecting the site's *Options* button (⋮) and clicking *Join*.

Members of a site can be given additional privileges within the site by using Liferay DXP's permission settings. It is also possible to assign different roles within the site to different members. This can be done through *site roles*, which are defined equally for all sites or *teams* which are unique for each site. These concepts will be discussed later in the chapter.

Liferay DXP separates site-scoped information from the Control Panel by placing it in the Sites menu. From this menu, you can select the specific site to work on. The Site Administration panel is available for your site, which includes Pages, Content, Members, Configuration, and Publishing.

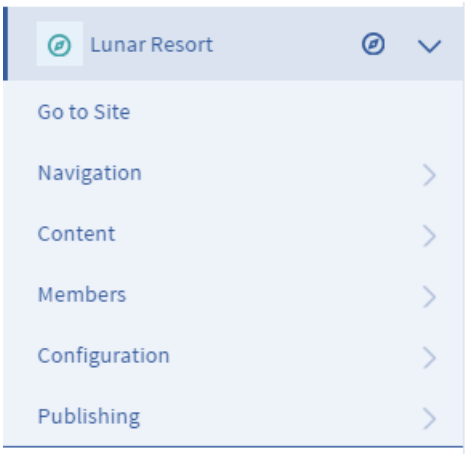


Figure 8.1: Your site's content resides in the Site Administration menu.

For details about Liferay DXP's social collaboration suite, see the Social Collaboration section.

Sites can also be organized hierarchically, just like organizations. The difference between sites and organizations, of course, is that sites are used to organize pages, content, application data, and users (via site memberships) whereas organizations are only used to group users. Content sharing is available for sites within the same hierarchy. For instance, if a parent site has a document called *Lunar Goals and Objectives* and would like for all its subsites to have a copy, the parent site's administrator can enable content sharing to automatically share the document with its subsites, instead of having to send each site the document individually. Also, content sharing privileges can be set to let every site administrator share content across sites they manage. Some examples of content you can share across site include web content structures and templates, categories, application display templates, etc.

Please refer to the Sites Admin Portlet section of Liferay's `portal.properties` file for a list of relevant configurable properties. For example, the `sites.content.sharing.with.children.enabled` property allows you to disable content sharing between sites and subsites, disable it by default while allowing site administrators to enable it per site, or to enable it by default while allowing administrators to disable it per site.

The Sites Directory application is a configurable app that can allow users to view a hierarchy of sites and subsites. It enables users to navigate to any of the displayed sites. To use this app to display site hierarchies, add it to a page, open its Configuration window, and under Display Style, select *List Hierarchy*. The My Sites Directory application is very similar to the Sites Directory application, except that it lists only the sites a user belongs to.

Each subsite in the hierarchy has its own administrator, and the Site Administrator role permissions do not flow down to child sites in the hierarchy. If a Site Administrator creates a subsite, he or she has the same permissions in that subsite. This is not, however, because of inheritance. It is only because creating a site

makes you the Owner of that site. A Site Administrator or a parent site has no default role in any subsites created by other Site Administrators.

If you wanted a user to have administrative access to all sites in a site/subsite hierarchy, you must create a role based on the Site Administrator role that has the permission *Manage Subsites*.

The Site Map application is another configurable app that's intended to help users navigate among pages within a site. When configuring this app, a site administrator can select a root page and a display depth. Just as sites can be organized hierarchically, so can the pages within a site. The display depth of the Site Map application determines how many levels of nested pages to display.

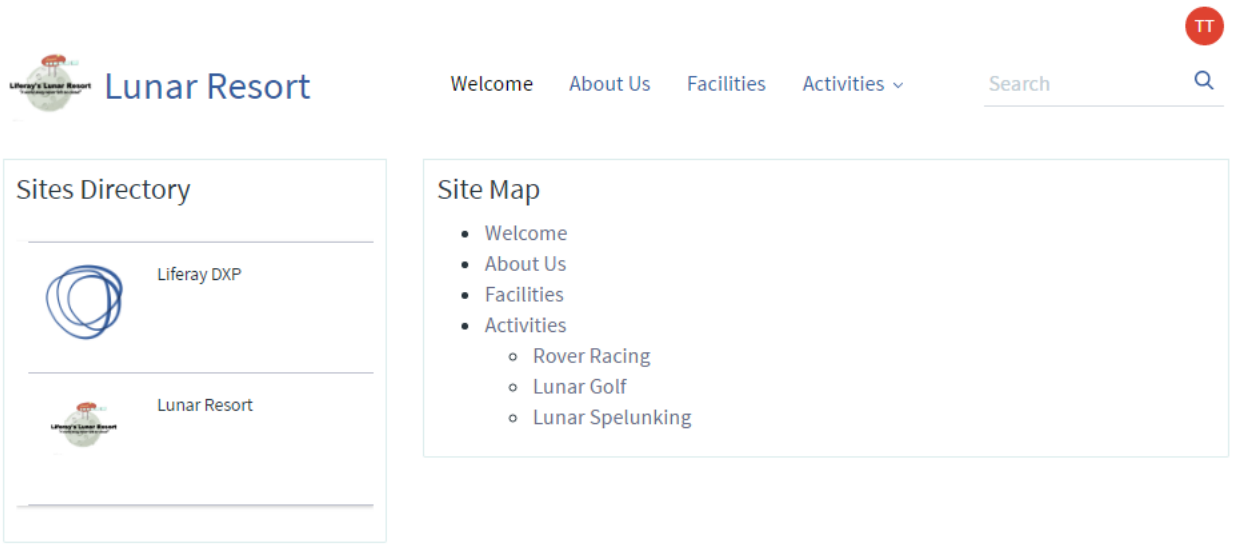


Figure 8.2: The Sites Directory application lets users navigate between sites organized hierarchically. The Site Map application lets users navigate among pages of a site organized hierarchically.

Another useful administrative application is the Site Members application. This enables administrators to survey all the users, organizations, and user groups that reside in the site. Similarly, Liferay provides the Portal Directory application, which functions the same as the Site Members app, but globally scoped for all sites in the instance.

Liferay DXP's sites have two categories of pages called page sets. There are two kinds of page sets: public pages and private pages. A site can have only public pages, only private pages, or both. Private pages can only be accessed by site members. Public pages can be accessed by anyone, including users who haven't logged in. It's possible to restrict access to pages at the page set level or at the level of individual pages through the permissions system. Public pages and private pages have different URLs and can have different content, applications, themes, and layouts.

Building a corporate intranet is a typical use case for Liferay sites. A corporate intranet could have sites for all the organizations in the company: Sales, Marketing, Information Technology, Human Resources and so on. But what about the corporate health and fitness center? That's something everybody in the company, regardless of organization, may want to join. This makes it a good candidate for an open and independent site. Similarly, the home page for a corporate intranet should probably be placed in an open independent site so any member of the instance can access it.

Tip: Prior to Liferay Portal 6.1, there were two ways of creating sites: organizations and communities. This has been simplified to provide more ease of use and allow for more flexibility. The main role of organizations is still to organize the users of the instance in a hierarchy but they can also have associated sites. Communities

can still be created through independent sites but the new name reflects the fact that sites can be used for many different purposes besides communities.

For other kinds of web sites, you may want to use independent sites to bring users together who share a common interest. If you were building a photo sharing website, you might have independent sites based on the types of photos people want to share. For example, those who enjoy taking pictures of landscapes could join a Landscapes site and those who enjoy taking pictures of sunsets could join a Sunsets site.

Liferay DXP always provides one default site, which is also known as the main site of the instance. This site does not have its own name but rather takes the name of the instance. By default the instance name is *Liferay* but this value can be changed through the simple configuration of the setup wizard. The instance name can also be changed at any time through the Control Panel within *Configuration* → Instance Settings*.

Adding Sites

Sites can be created through the Control Panel by a Liferay administrator. Liferay's Control Panel provides an administrative interface for managing your Liferay instance. There are four main sections of the Liferay Control Panel: Users, Sites, Apps, and Configuration. In this section, you'll learn how to use the Control Panel to manage sites. In a later section, you'll learn about using the Control Panel to manage site templates. Site templates allow Liferay administrators to create multiple sites with the same default set of pages and content. For information about the Apps, Users, and Configuration sections of the Control Panel, see the Leveraging the Liferay Marketplace, User Management, and Using the Control Panel sections, respectively.

Tip: If you're signed in as an administrator, you can access all sites by navigating to the Site Administration menu from the Control Panel. To manage a single site, navigate to the site by going to the Menu and clicking the *Site Selector* button ( ) from the sites dropdown menu and selecting the appropriate site name you'd like to manage. Once finished, the site administration options (i.e., Navigation, Content, Members, etc.) for that site are available.

To add a site for the Lunar Resort instance, navigate to the Control Panel and select *Sites* → *Sites*. Then click the Add icon ( ) at the bottom right of the page. If there is at least one site template available, a dropdown menu appears. Site templates provide a preconfigured set of pages, applications, and content that can be used as the basis of a site's public or private page set. To create a site from scratch, select *Blank Site*. Otherwise, select the name of the site template you'd like to use. If you opt to create a site from a site template, you have to choose whether to copy the site template's pages as your new site's public or private page set. If other site templates are created, they will appear in the Add menu as they become available. The following figure shows the form that needs to be filled when creating a *Blank Site*.

Name: names the site you wish to create. You also have the option to translate the name for many different languages. This can be done by selecting the language flag under the Name field, and inserting the name in the selected language. Liferay saves the name translation for each language and displays the translated site name when that specific language is selected for the instance. If a name translation is not provided, the default instance language's name is displayed.

Description: describes the site's intended function. The description can also be translated to other languages; see the Name description for more information on translating the site's description.

Active: determines whether a site is active or inactive. Inactive sites are inaccessible but can be activated whenever a site administrator wishes.

Membership Type: can be open, restricted, or private. An open site appears in the My Sites app and users can join and leave the site whenever they want. A restricted site is the same except users must request

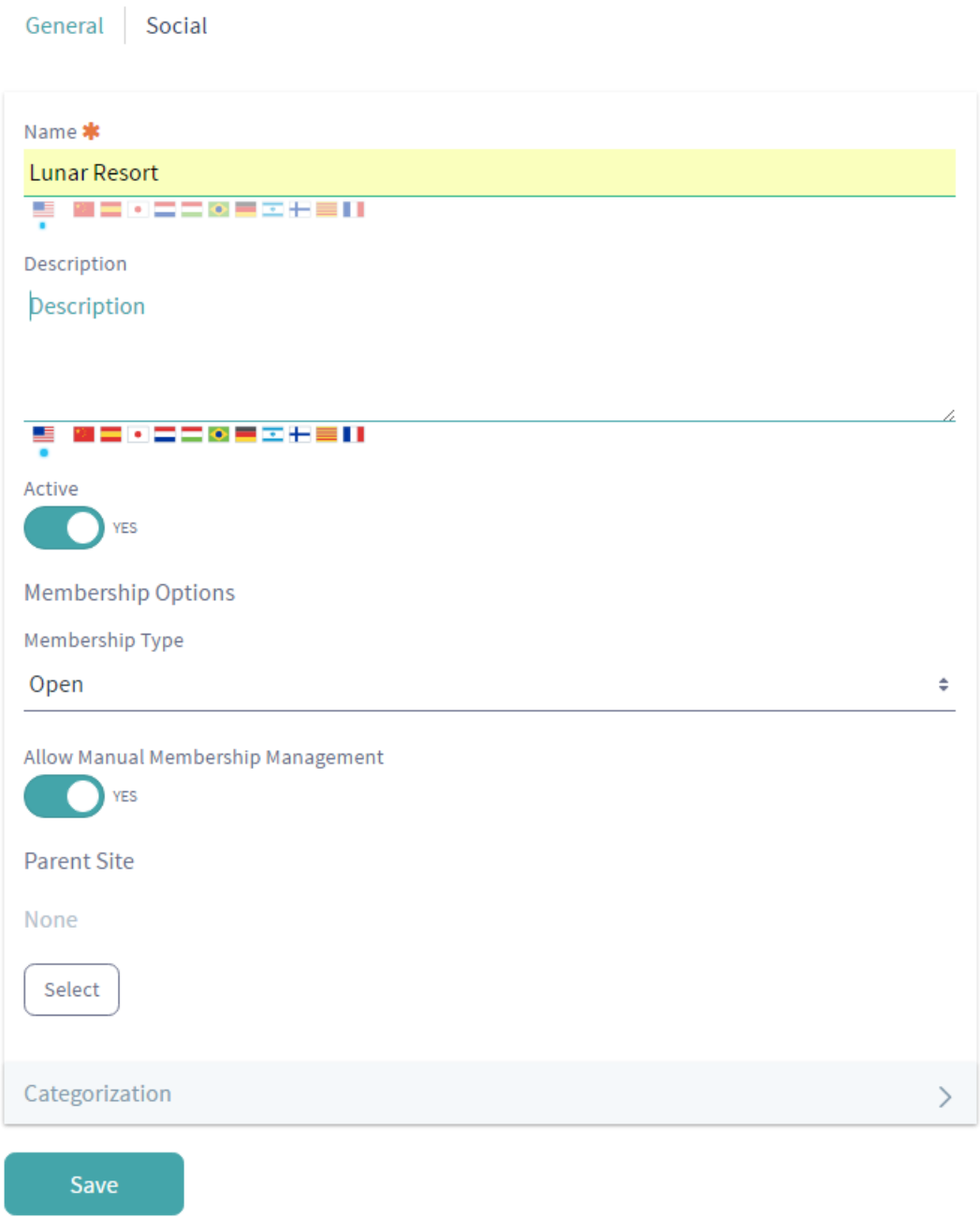


Figure 8.3: The New Site window aids in your new site development.

membership. A site administrator must then explicitly grant or deny users' requests to join. A private site does not appear in the My Sites app and users must be added to it manually by a site administrator.

Allow Manual Membership Management: determines whether to allow or disallow users to be manually added or removed from the site. By default, manual site membership management is enabled. This allows administrators to manually assign users to the site. It also allows users to join open sites or request membership from restricted sites using the My Sites app. For organization sites, manual site membership management is disabled, by default. This causes organization members to be automatically assigned membership following the organization's membership policy. Also, because manual membership management is disabled for organization sites, by default, the *Users* section of *Sites* is unavailable. To activate the *Users* functionality for your organization site, you'll need to check *Allow Manual Membership Management* after creating the organization site by navigating to its *Site Settings* menu.

Note: It's possible for site memberships to be handled automatically by a membership policy. The membership policy can check various pieces of information from each user, such as their first names, last names, birthdays, job titles, organizations, and user groups. Using this information, the site membership policy can automatically assign members to the site. If your site will implement a membership policy, your site administrators can disallow manual membership management for their site. When the Allow Manual Membership Management option is disabled, the *Members* section of Site Administration (Site Memberships and Site Teams) is hidden, even from administrators.

Parent Site: lets you select a parent site for the site that's being created. Sites can be organized hierarchically. Using hierarchical sites provides a simplified way to manage site memberships and site content sharing. For organizations that have attached sites, the organization hierarchy should match the site hierarchy. When you select a parent site, an additional option appears: *Limit membership to members of the parent site*. If this option is enabled, the site's membership policy performs a check so that you can only assign members to the current site if they're already members of the parent site.

When creating a blank site or organization site, the site is not immediately viewable. This is because sites without a page are impossible to view. Therefore, before you can view your site, you must first create a page for it. To add a page for your temporarily invisible site, navigate to the *Navigation* option from Site Administration. Then add a public page. After adding your site's first page, it renders and your site is viewable. For more information about adding pages, see the Creating and Managing Pages section.

You also have the option to categorize your site template using tags and categories by selecting the *Categorization* menu from the bottom of the page. To learn more about using tags and categories in Liferay, see the Organizing Content with Tags and Categories section. Lastly, you'll notice at the top of the page an additional tab named *Social*. This tab lets you manage whether users of your site can mention other users. You'll learn about mentioning users later in the Social Collaboration sections.

When creating a site from a site template, the initial form provides a new option that lets you decide if you want to copy the pages from the template as public pages or as private pages. By default, the site is linked to the site template and changes to the site template propagate to any site based on it. A checkbox appears that lets users unlink the site template if the user has permission to do so.

Once the site has been created, you'll want to configure its settings to fit your needs. A site's settings are broken into four categories: General, Social, Languages, and Advanced. You'll learn more about your site's settings in the next section.

Configuring Site Settings

You can access Site Settings by navigating to the Site Administation dropdown menu and selecting *Configuration → Site Settings*.

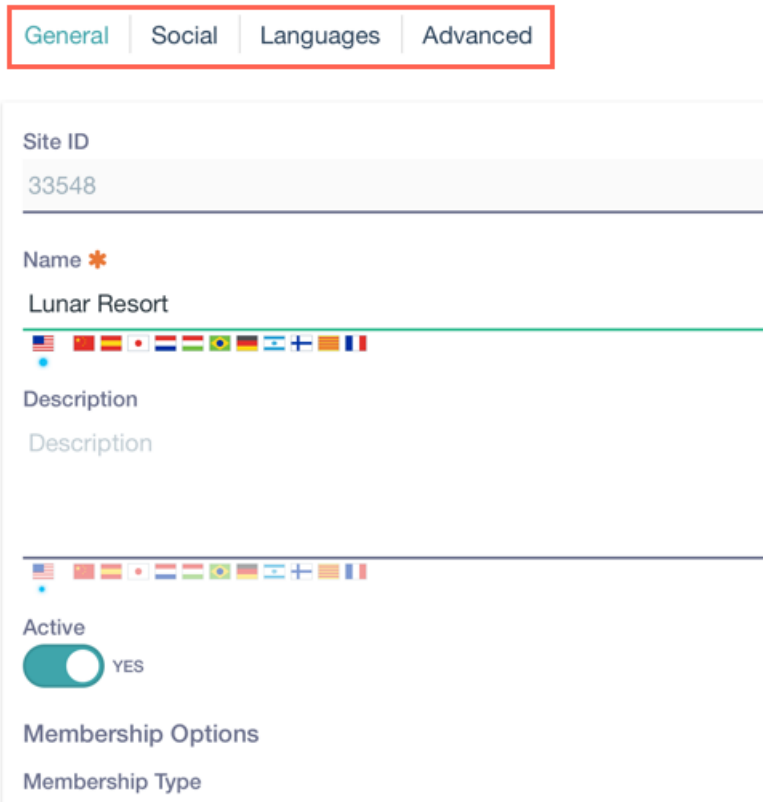


Figure 8.4: The Site Settings window offers a plethora of options for your site.

You'll find options to organize site content, manage site users, and improve your site's overall intelligence. You'll explore the Site Settings categories available, which are available from the *General*, *Social*, *Languages*, and *Advanced* tabs, and simulate configuring those settings for your sample Lunar Resort site.

Organizing Site Content Organizing your site's content drastically improves the usability of your site for users. The Site Settings menu offers some configuration options that aid in your organizational goals.

Categorization You can explore ways to tag and categorize your site by clicking the *Categorization* tab under General. These tools help administrators organize the site and allows for users to easily find your site and its content through search and navigation. To take full advantage of tags and categories, you'll need to add pages with content to your site. For more information on using tags and categories, visit the Organizing Content with Tags and Categories section.

Site Template The last tab listed under the Basic Information category is *Site Template*. If you created your Lunar Resort site as a Blank Site, this option is not available in Site Settings. If you did, however, create your site using a site template, this section displays information about the link between the site template and the site. Specifically, you can see which site template was used and whether or not it allows modifications to the pages inherited from it by site administrators. To learn more about site templates and how to create your own, see the Building Sites from Templates section.

Content Sharing If you select the *Content Sharing* tab from the Advanced tab, you can configure whether sub-sites can display content from this site. Administrators of this site's sub-sites can use all structures,

templates, categories, application display templates, etc. from this parent site. Even if you initially allowed content sharing between the parent site and its sub-sites, you're able to disable this option and immediately revoke content sharing from all sub-sites.

Recycle Bin The *Recycle Bin* option under the Advanced tab provides the option to enable/disable the Recycle Bin for your site. You can also regulate the age (in minutes) for which content is able to be stored in the Recycle Bin until it is permanently deleted. For a full explanation of the Recycle Bin, see the Restoring Deleted Assets section.

Custom Fields *Custom Fields* lets you edit the custom fields you already have configured for the Site resource. If you don't have any custom fields configured for the Site resource, you can navigate to the Control Panel → *Custom Fields* located under the *Configuration* tab. The Custom Fields tab does not display in Site Settings unless you have existing custom fields. For more information on Custom Fields, see the Custom Fields section.

Documents and Media The last option that relates to organizing your site's content is *Documents and Media*, which is found under General. This lets you enable/disable Directory Indexing, which allows site administrators to browse your site's documents and media files and folders. For example, a site administrator of a site called *Lunar Resort* can browse documents at http://localhost:8080/documents/lunar-resort if this option is enabled.

Now that you can organize your site's content using Site Settings, move on to the next section to learn how to manage site users.

Managing Site Users An always important job for site administrators is managing site users. There are configuration options in Site Settings that allow for easier user management.

Details *Details* is the leading option under the General tab, which provides the same menu you filled out when first creating your Lunar Resort site. This allows an administrator to change the description and membership type of a site. The membership type can be set as open, restricted, or private based on the privacy needs of the site. Users can join and leave an open site at will. To join a restricted site, a user has to be added by the site administrator. A user can also request to be added through the Sites section of the Control Panel. A private site is like a restricted site but doesn't appear in the Sites section of the Control Panel for users who aren't members.

You also have the ability to organize sites into hierarchies. At the bottom of the Details sub-section is the Parent Site section. This feature allows you to select the parent site for the site you're currently on. After selecting a parent site, you have a checkbox option to limit membership to members of the parent site.

Default User Associations Once you have the basic details of your site saved, you can begin assigning your users to roles and teams. *Default User Associations*, the leading option when opening the Advanced tab, lets you configure site roles and teams that newly assigned site members will have by default. If you'd like to learn more about creating roles and/or teams, visit the Roles and Permissions and Creating Teams for Advanced Site Membership Management sections, respectively.

Ratings To allow your site's users to rate content in your site, you can use the *Ratings* option to select what ratings type to use for applications like Documents and Media, Web Content, Comments, etc. Ratings types include Stars, Likes, and Thumbs. This is the leading option when opening the Social tab.

Mentions The last configuration option in Site Settings related to managing users is *Mentions*. *Mentions* is found under the Social tab. This option allows you to enable/disable the Mentioning functionality, which is used to *mention* (notify and/or draw attention to) friends and colleagues by entering the "@" character followed by their user name. You can learn more about the mentioning feature by visiting the Mentioning Users article.

Now that you're aware of your Site Settings abilities to improve User and Content management, you'll learn how to improve your site's overall intelligence using Site Settings.

Improving Site Intelligence Site Settings offers several miscellaneous configuration options that improve your site's usability, overall display, and data tracking. To make your Lunar Resort site smarter, explore and configure the options below.

Site URL One of the most important aspects of your site is the simplicity of your site URLs. Long obscure site URLs are a hassle for users to deal with, and can negatively affect your site's probability of being featured by search engines. Having a human-readable friendly URL assists indexing bots and is critical to good search engine optimization. For the Lunar Resort site, you can select the *Site URL* tab to set a friendly URL and/or a virtual host. The *Friendly URL* option lets you manage the path to your site in the instance's URL. Friendly URLs are used for both public and private pages. For public pages, the friendly URL is appended to http://localhost:8080/web. For private pages, the friendly URL is appended to http://localhost:8080/group. Each friendly URL needs to be a unique name, of course.

For example, suppose you want to set a friendly URL for your Lunar Resort site. If you set the friendly URL of your instance's default site to /lunar-resort, the URL of your default site's public home page would change to http://localhost:8080/web/lunar-resort/home. If your instance's default site had private pages, the URL of the default private home page would change to http://localhost:8080/group/lunar-resort/home.

Note that if you're adding a friendly URL for your instance's home page, you should update your instance's Home URL field so that page requests to http://localhost:8080 redirect properly. To do this, navigate to the *Configuration → Instance Settings* page of the Control Panel and find the Home URL field in the Navigation section. For the Lunar Resort example, you would enter */web/lunar-resort/home* into the Home URL field. Once you've entered this setting, page requests to localhost:8080 will redirect to the friendly URL of your Liferay instance's new homepage: http://localhost:8080/web/lunar-resort/home.

The other setting you can configure under the Site URL tab is *Virtual Hosts*, which makes web navigation much easier for your users by connecting a domain name to a site. This allows you to define a domain name (i.e., www.lunar-resort.com) for your site. This can be a full domain or a subdomain. This enables you to host a number of web sites as separate sites on one Liferay server.

For instance, if you set this up for the Lunar Resort's development network, users in that site could use *developers.lunar-resort.com* to get to their site, provided that the Lunar Resort instance's network administrators created the domain name and pointed it to the Liferay server.

To set this up, the DNS name *developers.lunar-resort.com* should point to your instance's IP address first. Then enter *http://developers.lunar-resort.com* in the Virtual Host tab for the Developers site. This helps users quickly access their site without having to recall an extended URL. The *Site URL* option is listed under the General tab.

Analytics Analyzing site traffic is another huge advantage for site administrators that want to monitor what content is most popular. Liferay DXP includes built-in support for Google Analytics, allowing administrators to make use of Google's tool set for analyzing site traffic data. When you sign up for Google Analytics, a snippet of code is provided which needs to be added to your web pages to allow Google's system to register

the page hit. It can be a tedious process to add this code to every page on a site, especially if it's a large site and there is a lot of user-generated content.

This problem can be solved in Liferay by putting Google's code into a custom theme written specifically for the web site on which Liferay is running. Doing this, however, requires a theme developer to make specific changes to the theme and it prevents users from using the many themes that are freely available for Liferay DXP "out of the box."

Because of this, support for Google Analytics has been built into Liferay DXP, and can be turned on through a simple user interface. This allows Liferay administrators to make use of Google Analytics on a site by site basis and turn it on and off when needed. You can sign up for Google Analytics at the Google Analytics site here:

http://www.google.com/analytics

To enable Google Analytics support, navigate to the *Analytics* tab in Site Settings, which loads a very simple form, pictured below.

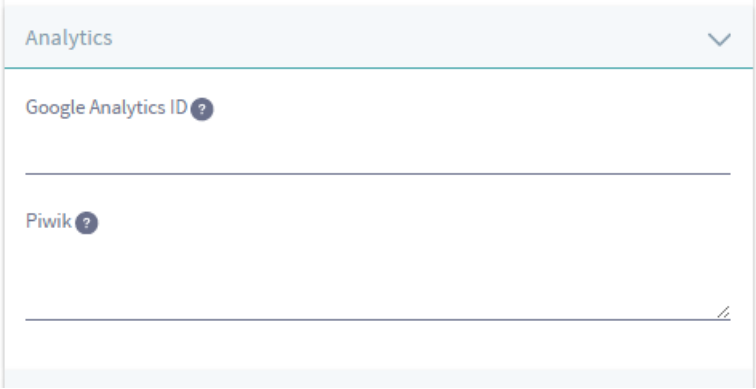


Figure 8.5: Setting up Google Analytics for your site is very easy: sign up for Google Analytics, receive an ID, and then enter it into the Google Analytics ID field.

Enter your Google Analytics ID (which should have been provided to you when you signed up for the service) in the field and click *Save*. All the pages in the site you selected will now have the Google Analytics code in them and will be tracked.

To enable a different analytics service, navigate to *Configuration* in the Control Panel, and then go to *Instance Settings → Miscellaneous*. You can enter the name of any additional service you want to add in the *Analytics* field provided. Once you have entered the name, go to the *Site Settings → Advanced → Analytics* page for the site where you wish to add analytics. Copy the JavaScript tracking code provided by your analytics platform into corresponding field for your service. Now all pages on the selected site contain the tracking script and will send analytics data to your analytics platform.

This is a fairly simple procedure, and it gives you the ability to take advantage of some great tools to help you visualize who's coming to your site and from where. The *Analytics* option is listed under the Advanced tab.

Maps The *Maps* options lets you configure the maps API provider used by your Liferay instance when displaying geolocalized assets. Geolocalized assets can be displayed for documents, web content articles, DDL records, etc. Maps is available under the Advanced tab.

Languages The *Languages* option lets you configure the language options for your site. This is the leading option when opening the Languages tab. You have options to use the default language options or define a custom default language.

Now that you know how to configure sites, you'll learn how to customize your personal sites.

Customizing Personal Sites

By default, newly created users in Liferay DXP are each granted a personal site. Each user functions as the site administrator of his or her personal site. Personal sites are fully customizable but cannot have more than one member. The public pages of personal sites provide a space for users to add content and applications that they'd like to make accessible to anyone, including guests. User blogs are often placed on public personal site pages. Content and applications that users would like to reserve for personal use are often placed on the private pages of personal sites. For example, each user can add a Documents and Media application to his or her private pages and use it as an online private file repository.

If you'd like to disable personal sites for your Liferay instance, just add the following properties to your `portal-ext.properties` file:

```
layout.user.public.layouts.enabled=false
layout.user.private.layouts.enabled=false
```

Note: The public and private page sets of personal sites are handled separately. You can leave one page set enabled while disabling the other.

What if you initially had user personal sites enabled for your instance but then disabled them? Each existing user's personal site remains on your Liferay instance until the next time they log in, at which point it's removed.

You can allow users to create personal sites but not have them automatically created for new users. To do this, first make sure that `layout.user.public.layouts.enabled` and `layout.user.private.layouts.enabled` are not set to `false`. You don't need to explicitly set them to `true`—`true` is the default. Then add the following properties to your `portal-ext.properties` file:

```
layout.user.public.layouts.auto.create=false
layout.user.private.layouts.auto.create=false
```

If the properties `layout.user.public.layouts.enabled`, `layout.user.private.layouts.enabled`, `layout.user.public.layouts.auto.create`, and `layout.user.private.layouts.auto.create` are all set to `true`, which is the default, then users will have personal sites and public and private pages will be automatically created for new users. There are a number of portal properties you can use to customize the automatically created pages. You can customize the names of the default pages, the applications that appear on the pages, the themes and layout templates of the default pages, and more. Please refer to the Default User Public Layouts and Default User Private Layouts sections of the `portal.properties` file for details.

Note: By default, users are able to modify the pages and applications of their personal sites. Administrators, however, can customize the modifiable portions of personal sites through Liferay DXP's permissions system by removing permissions from roles. To disallow all Liferay users from modifying something, remove the relevant permission from the User role.

Historically (prior to Liferay 5.1), only power users received personal sites. Back then, they were called personal communities. If you'd like only power users to receive personal sites, add the following properties to your `portal-ext.properties` file:

```
layout.user.public.layouts.power.user.required=true
layout.user.private.layouts.power.user.required=true
```

Personal sites are a dynamic feature of Liferay DXP. They allow users to manage and customize their own pages and content on your Liferay instance.

You've officially been introduced to Liferay's concept of sites. Using a point and click interface, you can create multiple web sites and define how users can access them, whether they are linked to a domain name, and create all of their pages. You can learn more about creating site pages in the next section.

8.2 Creating and Managing Pages

You've successfully created a site for your Liferay instance, but you may need to add new pages or edit existing pages. From the Site Administration → *Navigation* section of the Menu, your site pages can be accessed and configured. If you're not currently on the site you'd like to edit, click the *Site Selector* button ( ) next to your current site name in the Menu and select your desired site. The edits you make to your pages are only made to the scope you've selected in the Menu.

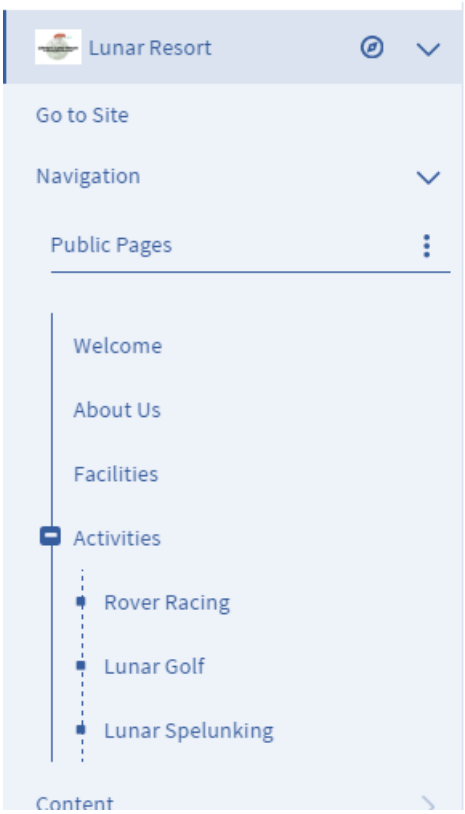


Figure 8.6: The Pages menu allows you to edit your site pages as a whole.

To add new pages to your site, click the *Options* icon ( ) for the page or page set you'd like to add a page to and select *Add Public Page*. Likewise, you can select *Add Private Page* to add a private site page. You can manage a plethora of options from the Navigation menu including page name, page layout, and page template.

You can also edit pages from the Navigation menu. To do this, select *Configure* from the Options icon next to a page or page set, which allows you to edit the page's basic information and configuration options.

In summary, the Navigation menu is used to navigate to pages, create new pages, and edit existing pages. Note that you can also switch to managing a set of pages in the Pages menu. Click on *Public Pages* or *Private*

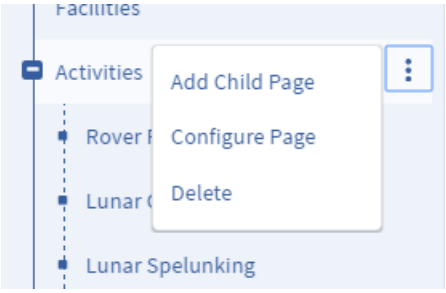


Figure 8.7: The *Options* button next to a page or page set allows you to add a child page, edit the existing page(s), or delete the page(s).

Pages (if private pages exist in your site) to manage that page group.

Liferay DXP's page groups are always associated with sites. Even users' personal pages are part of their personal sites. All pages belong to one of two types of page sets: public pages and private pages. By default, public pages are accessible to anyone, even non-logged in users (guests). Private pages are accessible only to users who are members of the site which owns the pages. This means the private pages of an organization's site would only be viewable by site members and members of the organization.

Regardless of whether the pages are public or private, Liferay DXP uses the same interface to manage them. You'll look at this interface more closely next.

Creating Pages

From the Navigation sub-menu in the main Menu, you can add a page or child page to any existing page or page set by clicking the *Add Page* button. Because *Public Pages* is selected on the left, clicking *Add Page* here adds a top level page making it a sibling page of the Welcome page. You can, however, nest pages as deeply as you like. To create a sub-page under the Welcome page, select the *Add Child Page* button next to *Welcome*.

The *Add Child Page* button lets you create child pages underneath the page you've selected. You can nest pages as deep as you like but for every page below the top level hierarchy you should provide navigation to it via a Navigation or Breadcrumb app, at least with most themes (including the default). Developers can create themes which have cascading menu bars which show the full hierarchy. Some examples of that are in Liferay's plugin repositories.

Note: The Navigation Menu app lets you configure a root page. This allows building navigation menus based on the page hierarchy under a specific page. You're given options in the Navigation Menu app's *Configuration* menu to specify the root page, root page level, and included pages.

Once you've selected the *Add Page* or *Add Child Page* button, you're directed to an *Add New Page* interface that lets you configure the page to your liking. If you later decide you don't like the order of your pages, you can drag and drop them in the list to put them in whatever order you want.

Note: You're not forced to define the page hierarchy in a page's friendly URL. Therefore, child page friendly URLs are not required to include their parent page. For example, a parent page named Parent and its child page named Child could have the URLs *SITE_URL/parent* and *SITE_URL/child*, respectively. The default friendly URL given to a page is based only on the page name and not the hierarchy. If you wish to modify a generated friendly URL, you can do so by following the Site URL configuration section.

Go ahead and add another top level page and name it *Community*.

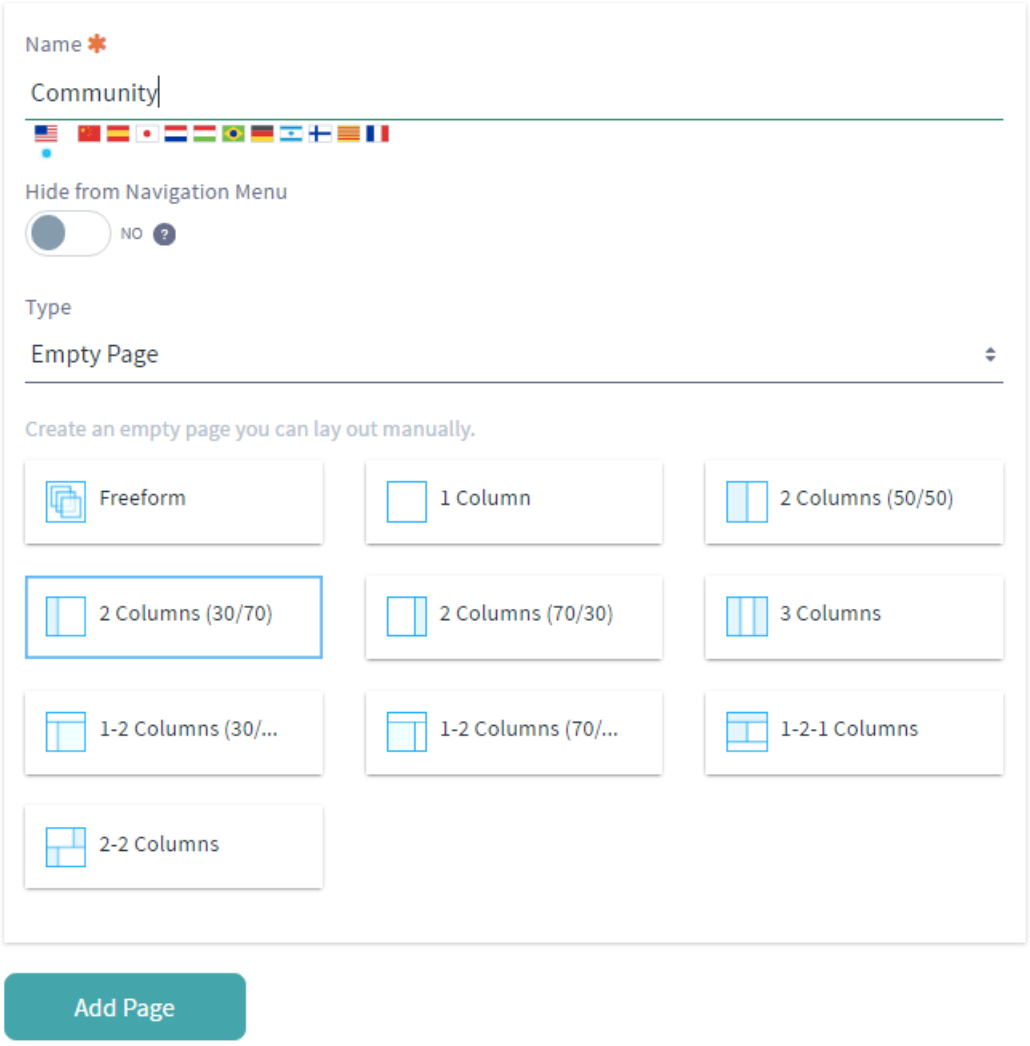


Figure 8.8: You can add a page to your site by giving it a name, page template, and page type.

When you create a new page, you can create either a blank page or a page prepopulated with apps from a page template. When you're entering the name of the page, you can select from a list of page templates that are currently available. You can learn more about Page Templates and how to use them in the Creating Pages from Templates section. To view the page you've added, click the page name from the left panel. By default, all pages are created as an empty page but in some situations, you might want to use one of the other options. You can take a look at all the page options you have below, along with their respective descriptions.

Empty Page: the default pages that are used. They have a layout which you can drag and drop portlets into. This page type creates an empty page with configurable column sizes.

Content Display Page: are used to search available content, explore related content with tags, and browse content catagories. This is a specialized page dedicated to managing web content from a page.

Blog: display content related to blogs. This is a specialized page dedicated to creating, editing, and viewing blogs.

Wiki: display content related to wikis. This is a specialized page dedicated to creating, managing, and viewing wiki articles from a page.

Page Set: exists as a container that holds child pages. These pages are not intended to hold content.

Because the page holds no content and has no purpose but to hold other child pages, there is no friendly URL for this page. It also cannot serve as a landing page for a site.

Panel: can have any number of portlets on them, as selected by an administrator, but only one will be displayed at a time. Users select which portlet they want to use from a menu on the left side of the page and the selected portlet takes up the entire page.

Link to URL: are redirects to any URL specified by an administrator. You can use URL pages to create links to pages belonging to other sites of your Liferay instance or to pages of an external site. Use URL pages cautiously since blind redirects create a poor user experience.

Embedded: display content from another website inside of your instance. An administrator can set a URL from in the page management interface and that page will appear in the context and within the navigation of your Liferay instance. To embed an external website, you must provide the protocol in the URL (e.g. `https://www.liferay.com/`).

Full Page Application: creates a page with one column that displays a single full page application.

Link to a Page of This Site: creates a page which functions as an immediate redirect to another page within the same site. You can select which page to link to from a dropdown in the page management interface. You could use a *Link to a Page of This Site* to place a deeply nested page in the primary navigation menu of your site, for example.

Copy of a Page of This Site: displays a copy of a pre-existing page in your Liferay instance.

Now that you know the basics of adding site pages, you can start working on the Lunar Resort site. If you're not currently on the site you're interested in adding pages to, navigate to Site Administration in the Menu, select the compass icon next to the current site name, and select the site you wish to edit.

If you're creating a site with many pages and sub-pages, the side panel view may seem constricted. In the Navigation menu, you have the option to expand the Pages view to allow for more room to work. Select the page set's *Options* icon and select *Expand* to see a full view of your page hierarchy.

As stated previously, if you ever need to modify the page you've created for your site, select the *Configure Page* button from the Options button from the Navigation menu in the Menu. When configuring a specific page, you're given more options than when you were creating a new page. These extended options for configuring specific pages are covered in more detail later in this section.

There are also configuration options that are only available for either indiviual pages or page groups only. You'll learn about options available for both instances.

You should be able to define and manage pages in Liferay at this point so next, you'll look at the options you have available for configuring your public or private pages as a whole, and the individual pages residing within those groups.

Customizing the Look and Feel of Site Pages When you select *Configure* for a page set, it defaults to the Look and Feel tab. On this tab, you're presented with an interface that allows you to choose a theme for the current site. Themes can transform the entire look of the portal. They are created by developers and are easily installed using the Liferay Marketplace. Since you don't have any themes beyond the default one installed yet, you'll use the default theme for your pages.

You can apply themes to individual pages as well by selecting the *Configure Page* option for a page and selecting the *Define a specific look and feel for this page* option under the *Look and Feel* category.

Many themes include more than one color scheme. This allows you to keep the existing look and feel while giving your site a different flavor. The *Color Schemes* option is not available for the default theme.

There are a few more configurable settings for your theme. You can switch the bullet style between dots and arrows and you can choose whether or not to show maximize/minimize application links by default. The *CSS* section allows you to enter custom CSS that will also be served up by your theme. In this way, you can tweak a theme in real time by adding new styles or overriding existing ones.

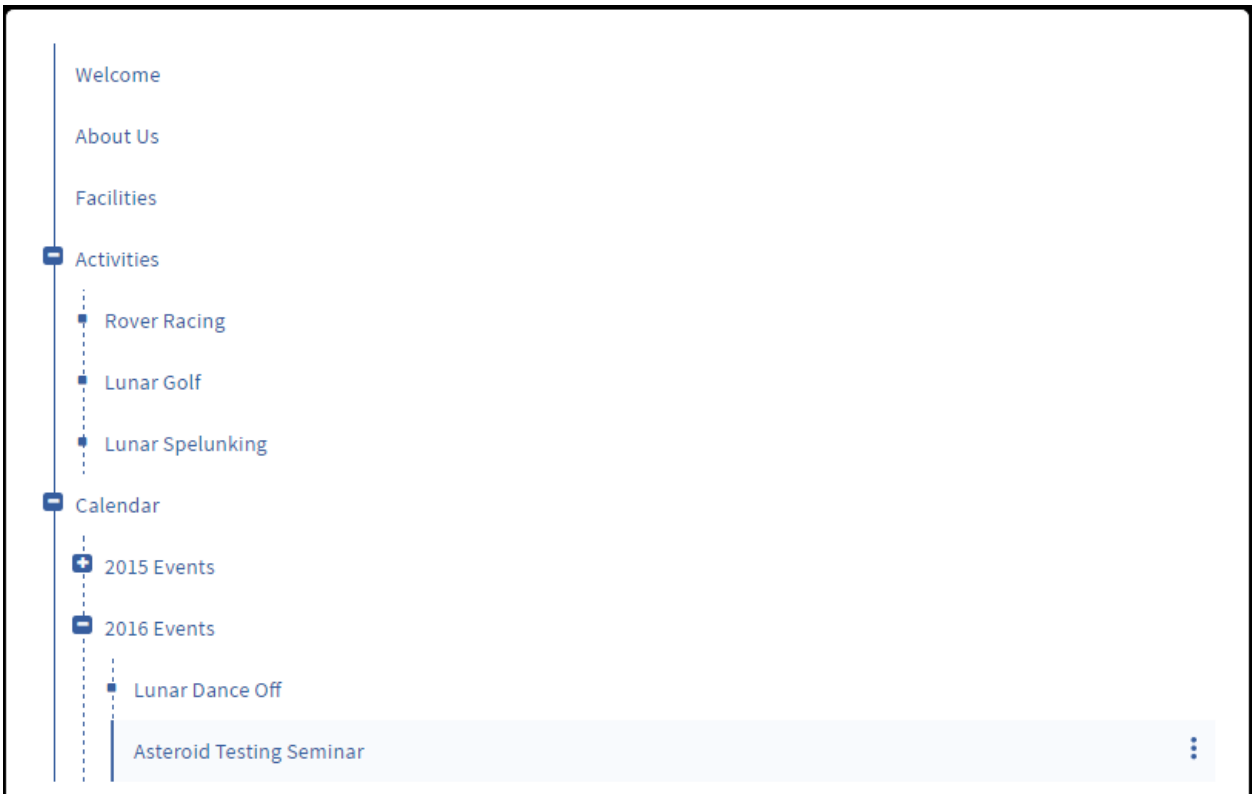


Figure 8.9: Expanding your page view gives you much more room to visualize your page hierarchy.

The next option configures the logo that appears for your site.

Using a Custom Logo for a Site By default, the Liferay logo is used for your site pages' logo. If you want to use your own logo for a specific site, use the *Logo* tab. Adding a custom logo is easy: select the *Logo* tab from the *Configure* interface and browse to the location of your logo. Make sure your logo fits the space in the top left corner of the theme you're using for your web site. If you don't, you could wind up with a site that's difficult to navigate, as other page elements are pushed aside to make way for the logo.

In the logo tab, you can also choose whether or not to display the site name on the site. If you check the box labeled *Show Site Name*, the site name will appear next to the logo. This option is enabled by default and cannot be disabled if the *Allow Site Administrators to set their own logo* option is disabled in *Instance Settings*. Removing the site name is not available for the default site – only newly created sites and user pages have the option to have the name display.

Executing JavaScript in Site Pages If you click on *Advanced* for a page set (either Public Pages or Private Pages), you'll find a window where you can enter JavaScript code that will be executed at the bottom of every page in the site. If your site's theme uses JavaScript (as is usually the case), it's best to add custom JavaScript code to the theme and *not* in this window. This way, all of your site's JavaScript code remains in one place.

Using the JavaScript window may be useful if your site's theme does *not* use JavaScript. In this case, the JavaScript window will contain *all* of your site's JavaScript and you can add some dynamic features to your site's pages.

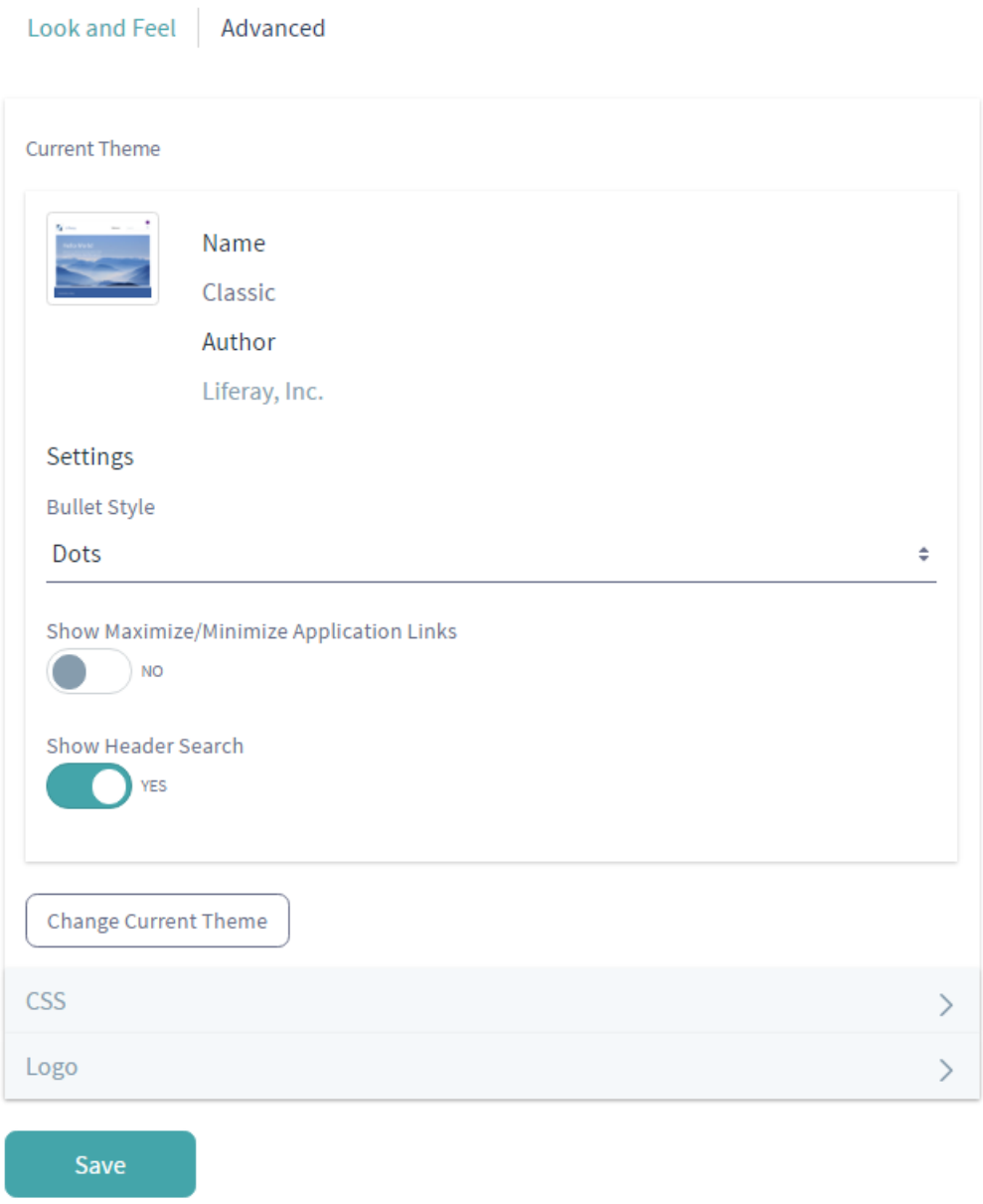


Figure 8.10: The Look and Feel interface allows you to choose a theme for the current site.

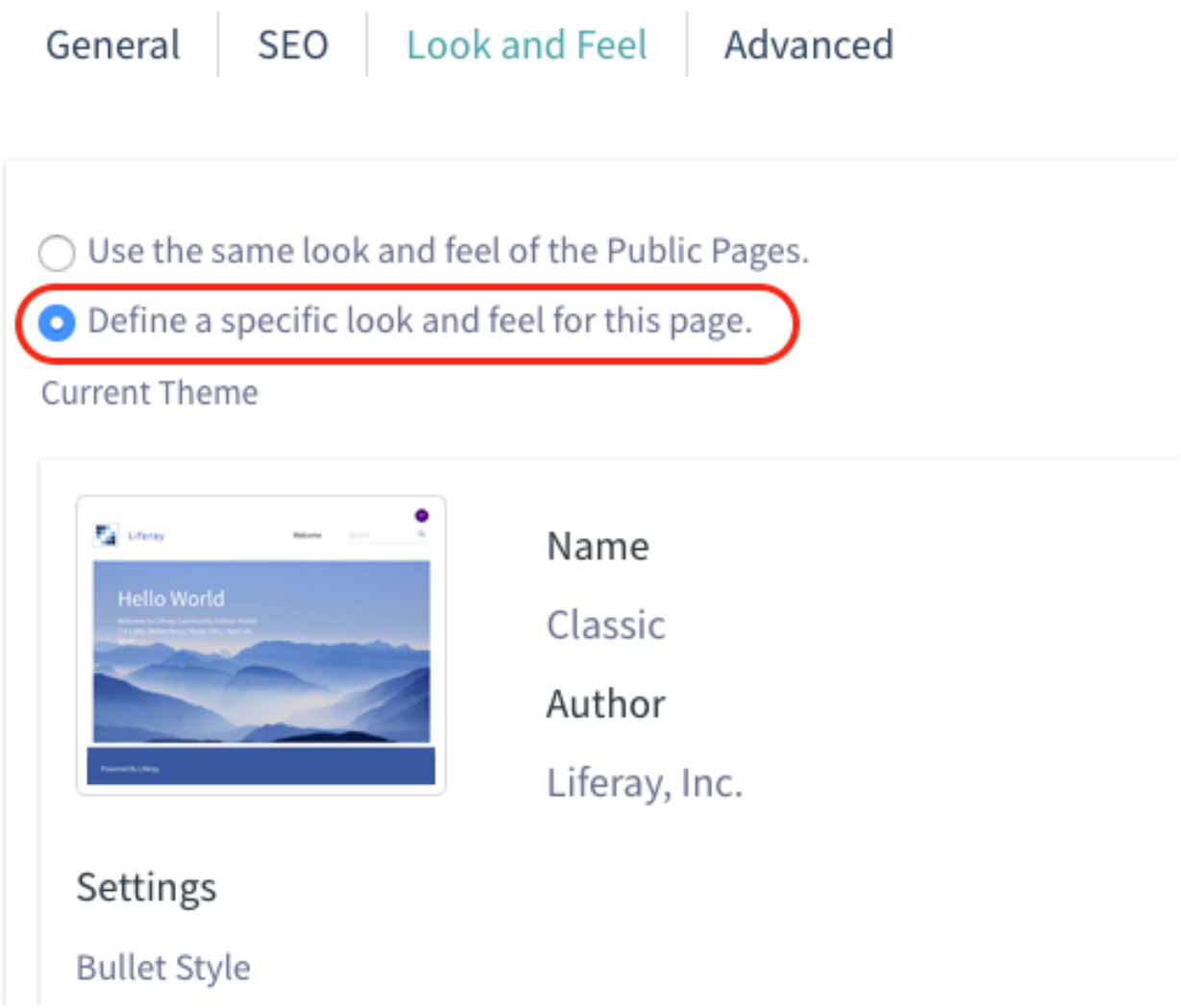


Figure 8.11: You can define a specific look and feel for a page.

Next, you'll look at an advanced features of the *Configure* interface: merging the current site's pages with the pages of the default site.

Merging Pages From Other Sites If you click on *Advanced* → *Advanced* from the Edit Public Pages interface, you'll find an option to merge the public pages of your instance's default site with the public pages of the current site. If you enable this option, the pages of the default site appear in the current site's navigation menu, along with the current site's pages. Also, the pages of the current site appear in the navigation menu of the default site, along with the default site's pages. This "merging" of pages only affects the list of pages in the default site's and the current site's navigation menus. This lets users more easily navigate from the current site to the default site, and vice versa. This option can be enabled for the public pages of both personal sites and regular sites.

Note that this merging of pages is not a "hard merge". For example, suppose that the site administrators of twenty different sites on your Liferay instance all enabled the *Merge Liferay public pages* option. Would the pages of all these different sites be merged into each site's navigation menu? No, that would make a mess!

Instead, Liferay DXP keeps track of the current `scopeGroupId` (the ID of the current site) and the previous `scopeGroupId` (the ID of the previously visited site). If the *Merge Liferay public pages* option is enabled for either the current site or the previous site, the pages of the default site are merged in the pages of the other site.

For example, suppose that your Liferay instance has three sites: the default site, site A, and site B. All three sites have some public pages. Site A has the *Merge Liferay public pages* option enabled; site B does not. When a user first logs in, he's directed to the default site. The `scopeGroupId` is that of the default site and there is no previous `scopeGroupId`, so no additional pages appear in the default site's navigation menu. Then suppose the user navigates to site A. Site A has the *Merge Liferay public pages* option enabled, so the default site's pages are added to site A's navigation menu. Now if the user goes back to the default site, site A becomes the previous site so site A's pages are added to the default site's navigation menu. If the user navigates to site B, no additional pages appear in site B's navigation menu because site B does not have the *Merge Liferay public pages* option enabled. And if the user navigates back to the default site, site B becomes the previous site, and, again, since site B does not have the *Merge Liferay public pages* option enabled, no additional pages are added to the default site's navigation menu.

Rendering Pages for Mobile Devices You can configure your page set for mobile devices by selecting the *Advanced* tab and selecting *Mobile Device Rules* option. The ability to modify themes for regular browsers and mobile devices can only be accomplished by using this option. Mobile device rules are inherited from your Public Pages, but you can define specific rules per page. With the ability to define different rules per page, you can edit the Look and Feel of specific pages for mobile devices, including the theme.

Configuring Rules for Virtual Hosting If you're using virtual hosting for this site, you can configure `robots.txt` rules for the domain by selecting the *Robots* option from the *Advanced* tab. The Robots page gives you the option to configure your `robots.txt` for both public and private pages on a site. If you don't have Virtual Hosting set up, this tab is rather boring.

Notifying Search Engines of Site Pages If you select the *Sitemap* option from the *Advanced* tab for a page set, you can send a sitemap to some search engines so they can crawl your site. It uses the sitemap protocol, which is an industry standard. You can publish your site to Yahoo or Google and their web crawlers will use the sitemap to index your site. Liferay DXP makes this very simple for administrators by generating the sitemap XML for all public web sites.

By selecting one of the search engine links, the sitemap will be sent to them. It's only necessary to do this once per site. The search engine crawler will periodically crawl the sitemap once you've made the initial request.

If you're interested in seeing what is being sent to the search engines, select the *preview* link to see the generated XML.

Next, you'll learn how to customize individual site pages.

Customizing Pages

When you decide to customize a single page, some different options that were not available when initially creating a page appear. Customizing a specific page can be done by navigating to the Navigation menu in the main Menu and selecting the *Options* icon next to the specific page you'd like to edit from the navigation tree. From the Options dropdown, select *Configure Page*. There are three groups that the options for customizing a specific page can be defined under. You'll learn what each group offers for your site pages.

Managing Page Content Managing your page's content drastically improves your page's organization and user experience. The site page's configuration options offers some oppurtunities to organize page content for your Lunar Resort site.

Categorization You can explore ways to tag and categorize your page by clicking the *SEO* tab, which shows how the categorization options by default. These tools help administrators organize the page and allows for users to easily find your page and its content through search and navigation. For more information on using tags and categories, visit the Organizing Content with Tags and Categories section.

Custom Fields *Custom Fields* lets you edit the custom fields you already have configured for the *Page* resource, which is accessible from the *Advanced* tab. If you don't have any custom fields configured in your site, this option is not available. If you don't have any custom fields configured for the Page resource, you can navigate to the Control Panel → *Custom Fields* located under the *Configuration* tab. These are metadata about the page and can be anything you like, such as author or creation date. For more information on Custom Fields, see the Custom Fields section.

Improving Page Usability One of the most important tasks for administrators is ensuring the user experience is easy and enjoyable. There are configuration options that aid in providing a seamless user experience for your site page.

Details The first option you're given (and the default option selected when customizing an individual page) is *Details*. This option lets you name the page for any localizations you need, set whether the page is hidden on the navigation menu, set an easy to remember, friendly URL for the page, and select the page type. Plus you can specify how apps are arranged on a page. Choose from the available installed templates to modify the layout.

Note: If you require a more complex page layout, you can install the Liferay Layout Templates app from Liferay Marketplace to access four additional layout templates.

It's easy for developers to define custom layouts and add them to the list. This is covered more thoroughly in the tutorial Layout Templates with the Liferay Theme Generator.

Look and Feel The *Look and Feel* option lets you set a page-specific theme. You can inherit what you already have configured for your page sets' theme, or you can uniquely define them per page. You can see the Page Set's Customizing the Look and Feel of Site Pages section for more details.

Mobile Device Rules This option allows you to apply rules for how this page should be rendered for various mobile devices. You can set these up by navigating to your Site Administration dropdown menu and selecting *Configuration* → *Mobile Device Families*.

Embedded Portlets This option only appears if you have embedded one or more portlets on the page. Previous to Liferay DXP 7.0, you were able to embed portlets on a page by acquiring their portlet ID and using a `runtime-portlet` tag in the Web Content Display app to embed a portlet on a page. This is no longer possible.

Applications can now be embedded on a page via web content template. To learn more about this, see the Adding Templates section. Also, you can embed a portlet in themes programmatically. If you're interested in learning more about this, visit the Embedding Portlets in Themes tutorial.

Customization Settings This configuration option located in the *Advanced* tab lets you mark specific sections of the page you want users to be able to customize. You can learn more about page customizations in the Personalizing Pages section.

Enhancing Page Intelligence Your page's configuration options offers several opportunities to improve your page's data optimization and JavaScript, among others. To enhance your page's intelligence, configure the options described below.

SEO *SEO* provides several means of optimizing the data the page provides to an indexer that's crawling the page, which is accessible from the *SEO* tab. You can set the various meta tags for description, keywords and robots. There's also a separate Robots section that lets you tell indexing robots how frequently the page is updated and how it should be prioritized. If the page is localized, you can select a box to make Liferay DXP generate canonical links by language. If you want to set some of these settings for the entire site, you can specify them from the Sitemaps and Robots tabs of the Manage Site Settings dialog box (see below).

In previous versions of Liferay, it was possible that a single page could be indexed multiple times. In Liferay Portal 6.1, all URLs that direct to the same page will only create one entry in the index. Previously, the simple URL *http://www.lunar-resort.com/web/guest/blog/-/blogs/themoon* and different versions of the URL which provided additional information about the referring page had different entries in the index. In Liferay DXP 7.0, each asset (web content article, blog entry, etc.) has a unique URL. From the search engine's point of view, this will make your pages rank higher since any references to variations of a specific URL will all be considered references to the same page.

JavaScript If you click on *Advanced* → *JavaScript*, you'll find a window where you can enter JavaScript code that will be executed at the bottom of your page. If your site's theme uses JavaScript (as is usually the case), it's best to add custom JavaScript code to the theme and *not* in this window. This way, all of your site's JavaScript code remains in one place.

This configuration option is also available for page sets like Public Pages and Private Pages. Visit the Executing JavaScript in Site Pages section for more information on doing this for page sets.

Advanced The *Advanced* option contains several optional features. You can set a query string to provide parameters to the page. This can become useful to web content templates, which you'll see in the next chapter. You can set a target for the page so that it either pops up in a particularly named window or appears in a frameset. And you can set an icon for the page that appears in the navigation menu.

Next you'll learn how to add applications to a site page.

Adding Applications to a Page

Liferay DXP pages are composed of applications. All of your site's functionality, from blogs to shopping, is composed of apps. Even static web content can be displayed through Web Content Display apps. To add an app to a page, just click the *Add* button ( ) from the top menu and select the *Applications* tab. You can either browse through the categories of available apps until you find the one you're looking for or you can search for apps by name. Once you've found an app, click the *Add* button to add it to the current page. Once it's been added to the page, you can drag it to a new position. Alternatively, you can drag the app directly from the Applications menu to a specific location on the page. Follow the steps below to add some Collaboration apps to the Lunar Resort site.

1. From the top menu, select *Add* → *Applications*.
2. In the menu that appears, expand the *Collaboration* category.

3. Drag the *Blogs Aggregator* app from the Add Application window to the right column of your page.
4. Next, drag the *Wiki* app to the left column.

See how easy it is to add applications to your pages? You've added the Wiki app and Blogs Aggregator app to a page.

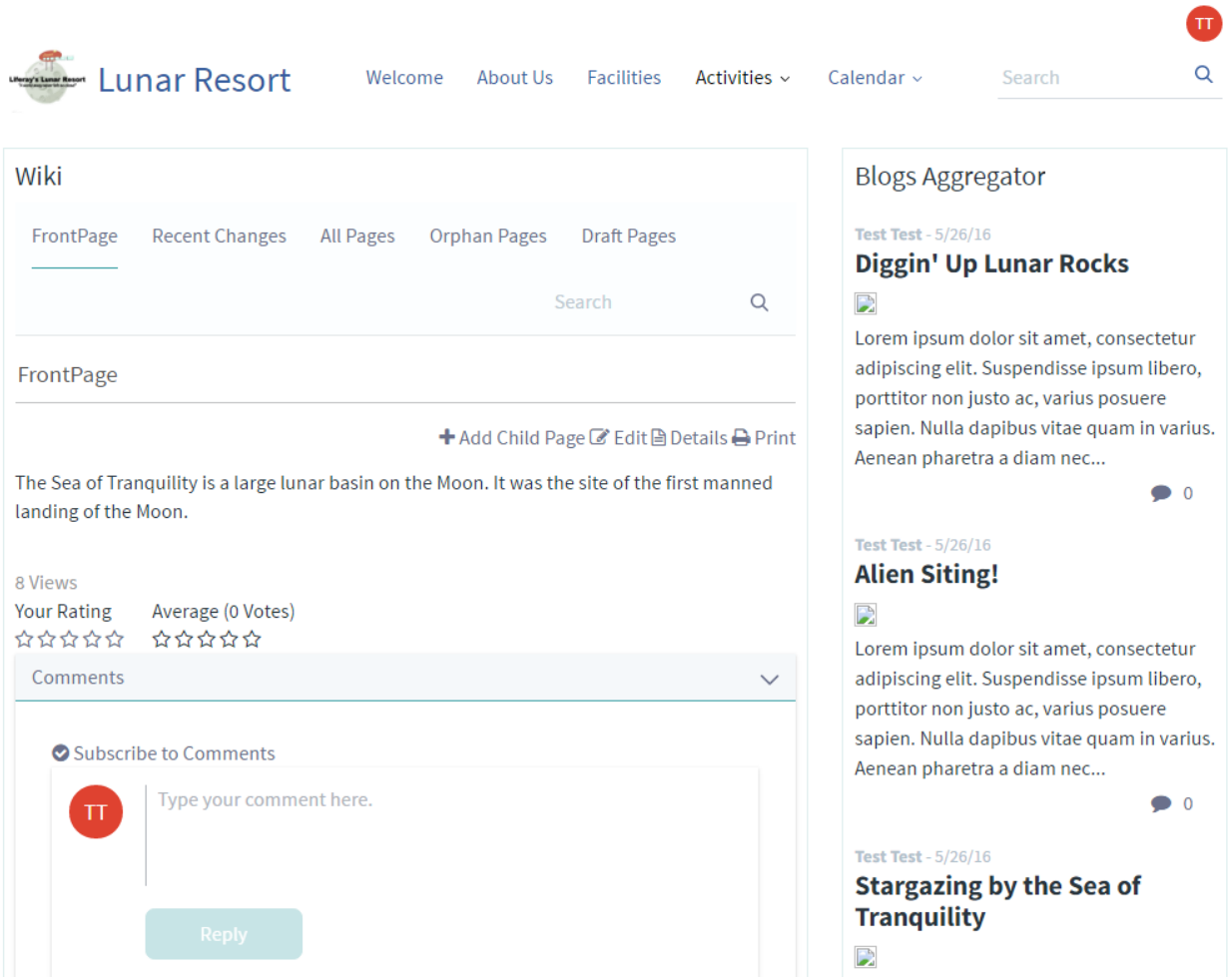


Figure 8.12: Your page layout options are virtually limitless with a slew of application and layout combinations.

It's easy to make your pages look exactly the way you want them to. If the default layout options provided aren't enough, you can even develop your own. For more information about developing custom layout templates, see the tutorial Layout Templates with the Liferay Theme Generator.

Next, you'll practice personalizing pages using page customizations!

Personalizing Pages

Administrators can designate pages or sections of pages to be customizable. When a user visits such a page, a notification appears stating that the user can customize the page. Users can make customizations only in the sections of pages designated by administrators. Customizations are based on the rows and columns of a page layout. Page customizations are only visible to the user who made the customizations. By default, site members can make page customizations but non-site members and guests can't.

To enable page customizations as an administrator, first click *Configure Page* from the *Options* button next to the Page you'd like to let site members modify. Then select the *Advanced* tab at the top of the page, expand the *Customization Settings* area, and click on the *Customizable* selector button.

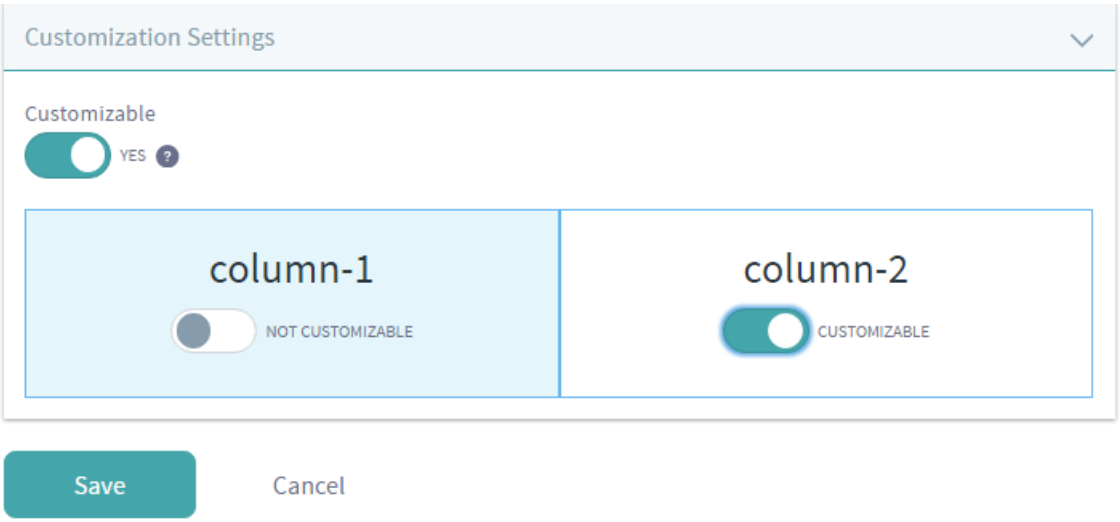


Figure 8.13: To enable page customizations, click on the *Configure Page* button next to the page, expand the *Customization Settings* area, and click on the *Customizable* button.

Once you've enabled the *Customizable* selector, you can select the sections of the page you'd like to enable customization for, depending on the layout template of your page. Enable one or more of the *Customizable* sections to allow site members to customize certain sections of the page. Regions that you've designated as customizable are colored blue.

When site members visit your customizable page, they'll see an extended Control Menu with a notification saying *You can customize this page*. Site members can toggle whether to view or hide the customizable regions. If you toggle the selector to view customizable regions, the regions on the page are color-coded to help distinguish customizable vs. non-customizable sections of the page.

Site members can also choose between viewing their customized page and viewing the default page by selecting the *Options* button ( ) from the Control Menu and clicking the *View Page without my customizations* or *View My Customized Page*.

There's also a *Reset My Customizations* option available from the *Options* button that restores a user's customized page to match the default page. This allows users to discard one set of customizations and start a new set without having to manually undo each customization that they'd previously made.

Note that non-administrator site members can access the Add menu from the top right side of the screen when viewing their customizable page even if they don't ordinarily have permission to view this menu. This allows them to add apps to the sections of the page that they're allowed to customize. If they click *View Page without my customizations*, the Add menu will disappear from the menu since they're not allowed to modify the default page.

Administrators of customizable pages have the same two views as site members: the *default page* view and the *customized page* view. Changes made to the default page affect all users, whereas changes made to the customized page affect only the administrator who made the changes. Changes made by administrators to non-customizable sections in the default view are immediately applied for all users. Changes made by administrators to customizable sections, however, do *not* overwrite users' customizations.

Users can make two kinds of customizations to customizable regions. First, they can configure applications within the customizable regions. Second, they can add apps to or remove apps from the customizable

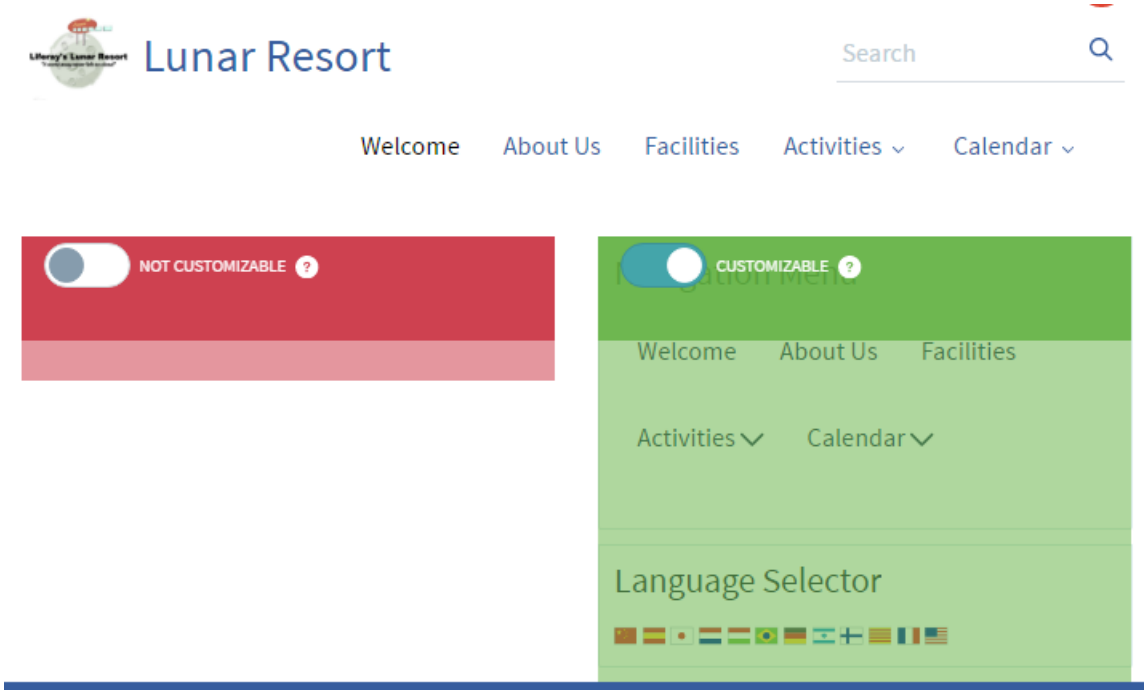


Figure 8.14: Customizable regions are colored green and non-customizable regions are colored red.

regions.

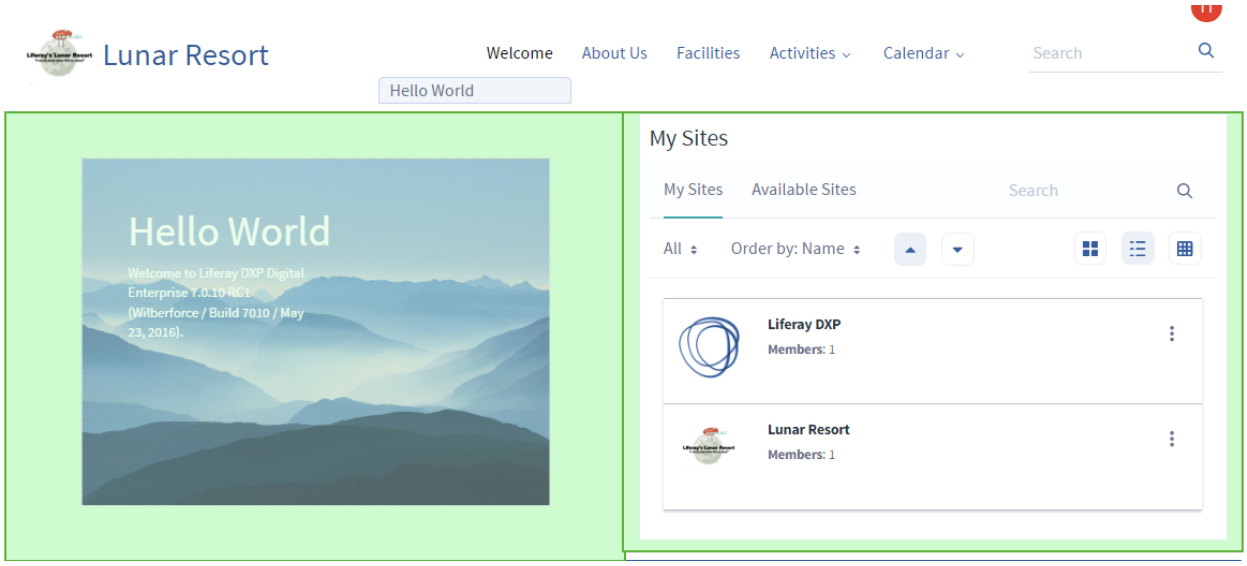


Figure 8.15: : Customizable areas are highlighted green when organizing apps on the page.

Liferay DXP doesn't allow users to change a non-instanceable app's configuration inside a customizable region since those kinds of apps are tied to the site to which they've been added. If this were allowed, the customization would affect all users, not just the one who customized the region. Therefore, changes to the app configuration in a customizable region are only possible for instanceable apps, whose app configuration only affects that one user.

For example, suppose that you, as an administrator, selected the right column of the Welcome page of the Lunar Resort site to be customizable. A member of the Lunar Resort site could take the following steps to make a personal customization of the Welcome page:

1. Navigate to the Welcome homepage by clicking the Site Administration's *Site Selector* button ( ) and selecting the Lunar Resort site.
2. Add the Language Selector app to the right column of the page by clicking the *Add* icon, clicking on *Applications*, searching for *Language Selector*, and clicking *Add* next to its name.

The Language Selector application is useful to have on your homepage if you expect users who speak different languages to access your instance. Users can select their language in the Language Selector app to view a translation of your site into their native language. After closing the Configuration dialog box of the Language Selector app, the customized Welcome page looks like this:

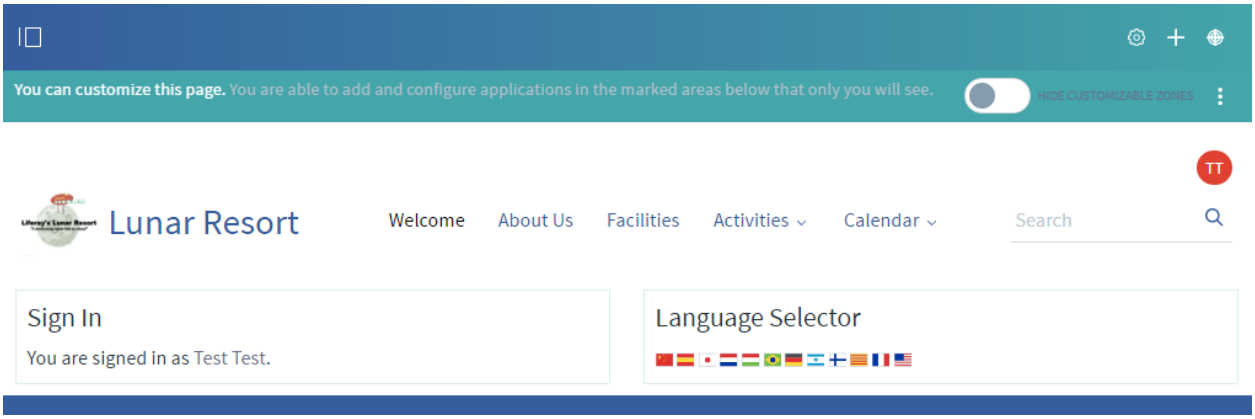


Figure 8.16: : In this example, the user added the Language app, and changed the display style from icons to a select box.

To allow users to customize a page, administrators must grant users permission to *Customize* pages under the Site section. This can be achieved by assigning permission to a role, then assigning this role to the appropriate users. For example, if you want any logged user to be able to customize your customizable pages, you could assign the *Customize* permission to the role *User*. If you want site members to be able to customize the customizable pages of their sites, you would accept the default setting. By default, the *Customize* permission is assigned to the role *Site Member*.

In addition to granting the ability to customize app configurations, the *Customize* permission allows users to customize the look and feel of apps and to import or export app settings. Next, you'll look at how to change page permissions.

Changing Page Permissions

By default, public pages are just that: public. They can be viewed by anybody, logged in or not logged in. And private pages are really only private from non-members of the site. If someone has joined your site or is a member of your organization, that person can see all the private pages. You can, however, modify the permissions on individual pages in either page group so only certain users can view them.

Suppose you want to create a page only for administrators to see. You can do this with the following procedure:

1. Go to your site's Site Administration dropdown and select *Navigation → Private Pages*. If you don't have the *Private Pages* option available, select the *Options* button next to Public Pages and click *Add Private Page*. Remember, these pages by default are viewable only by members of the site.
2. Create a page called *Admin Tips*.
3. Click *Configure Page* from the Options button dropdown for the page in the left menu.
4. Select *Permissions* from the *Options* icon ( ) in the top right corner of the screen.
5. Uncheck the *View* and *Add Discussion* permissions next to the Site Member role.
6. Click the *Save* button.

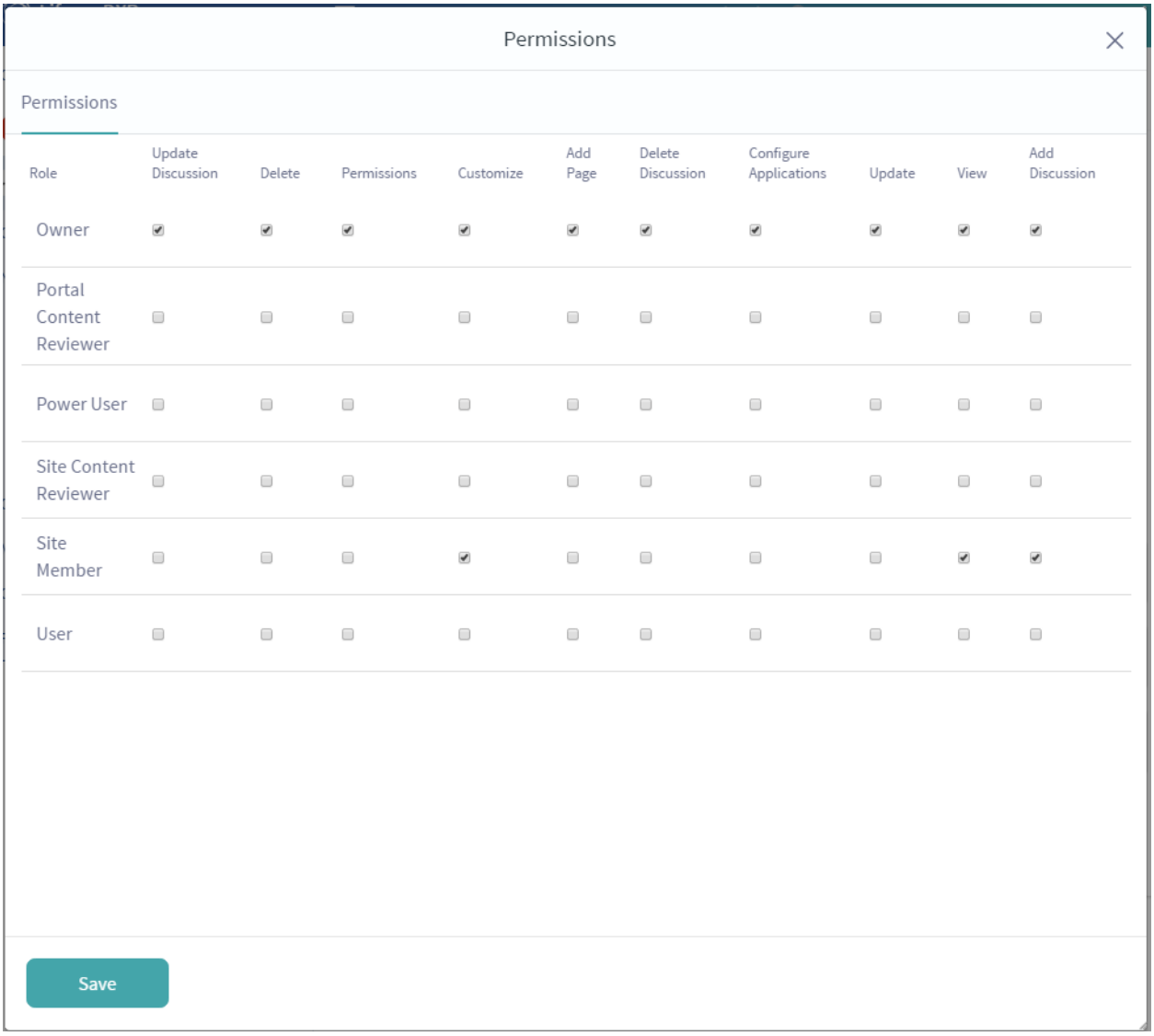


Figure 8.17: : The Permissions offer a plethora of options for each role.

Congratulations! You've changed the permissions for this page so only site administrators can view it. Any users you add to this role can now see the page. Other users, even members of this site, won't have permission to see it.

Pages in Liferay DXP are as flexible as pages you'd create manually without a Liferay instance. Using a point and click interface, you can define your site any way you want. You can create and remove pages, export and import them, set their layouts, define how they are indexed by search engines, and more.

You now understand how to manage pages in Liferay DXP. It's time to move on to further customizing those pages for mobile devices and building standardized pages using custom templates.

8.3 Displaying Pages on Mobile Devices

Mobile device families allow you to configure sets of rules to alter the behavior of the instance based on the device being used to access Liferay DXP. The proportion of mobile device users browsing the web has been steadily increasing, so it's important to be able to handle different kinds of devices appropriately. For instance, you can configure the look and feel of Liferay pages accessed by smartphone or tablet users differently from those accessed by PC users.

Both sites and individual pages can be configured with any number of mobile device families. A family is designed to describe a group of devices. It can contain one or more rules that describe a category of devices, such as all Android devices or all iOS tablets. You can define as many rules in a family as you need to classify all the devices for which you'd like to define actions. Families can be prioritized to determine which one applies to a given page request.

In order to configure mobile device rules, you need a way to find out the characteristics of the device. While some of the characteristics are provided by the device, most are not. For this reason, there are databases that contain information about thousands of devices. These databases make it possible to learn every detail about a device from the device type, which is included in each request sent to Liferay. Liferay DXP's Mobile Device Rules can connect to device databases so that you can use their device characteristics in your rules.

Important: For the features described in this article to work, you must install the Liferay Mobile Device Detection (LMDD) app from the Liferay Marketplace. This app provides the device detection database that's required for your Liferay DXP instance to detect which mobile devices are accessing it. Note that if you're running Liferay DXP, you must install the lite version of LMDD before you can install the enterprise version. Click here for instructions on using Liferay Marketplace to find and install apps.

It's possible to develop plugins that integrate with other device databases. Even if you don't have a device database, you can still set up mobile device rules. They won't, however, be effective until a database is deployed, because the portal won't have enough information about the devices being used to make page requests. To learn how to tap into Liferay DXP's Device API, see the Using the Device Recognition API tutorial.

You can access the Mobile Device Families administrative page from the Configuration section of Site Administration. Make sure you're on the appropriate site before adding mobile device families via Site Administration. You can also add families for all sites by navigating to the Control Panel → *Sites* → *Global*. The Mobile Device Families administrative page displays a list of defined families and lets you add more. To add rules to a family, select the device family name and select the *Add* button ().

The rules defined for a family, along with the priorities of the families selected for a particular site or page, determine which family's actions are applied to a given request. From the New Classification Rule page for a specific rule set, you can add a rule by specifying an operating system, rule type, physical screen size, and screen resolution. Remember that you can add as many rules to a family as you need in order to classify the devices on which you'd like to take actions. You'll notice after saving the classification rule that it's characterized as a *Simple Rule*. By default, only the Simple Rule type is available. The rules are designed to be extensible, and additional rule types can be added by your developers.

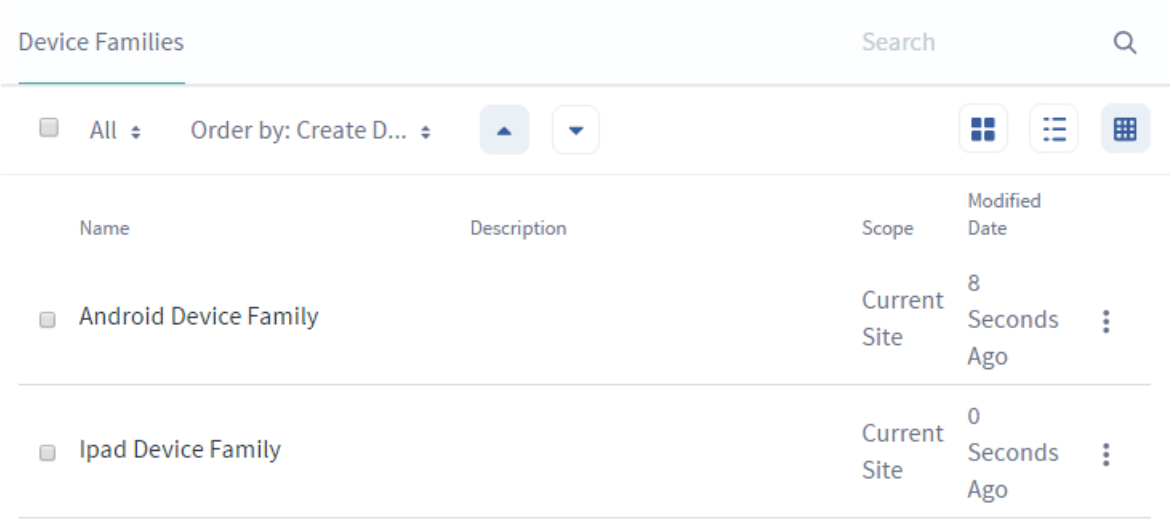


Figure 8.18: You can manage device rules by clicking on a Mobile Device Family.

Once you've created some mobile device families and added some rules to them, you'll be ready to set up some actions. The actions defined for a family determine what happens to a particular request when the device is detected and the family has been found to apply.

Tip: The Audience Targeting application offers a *Device* rule that evaluates whether the user is accessing content using a particular device family. This rule is integrated with the Mobile Device Families app. Visit the Liferay Audience Targeting Rules section for more details.

You can add actions to a family from the Navigation menu of Site Administration. Select a page set's (e.g., Public Pages) *Options* ( ) → *Configure* button and then select the *Advanced* tab and the *Mobile Device Rules* option in the bottom menu. Use the *Select* button to select families to be applied either to a page group or to a single page. If you select the page group itself from the left-hand menu, the selected family applies to all the pages of the site by default. If, however, you select an individual page and then click the *Select* button, the families apply only to that page. You can select multiple families for a particular site or page and order them by priority. The families are checked in decreasing order of priority: the actions defined by the first family that applies are executed.

To add actions to a selected rule group, use the *Actions* ( ) → *Manage Actions* button and then click *Add Action*. By default, there are four kinds of actions that can be configured for mobile families: layout template modifications, theme modifications, URL redirects, and site redirects. Layout template modifications let you change the way portlets are arranged on pages delivered to mobile devices, and themes modifications let you select a specific look and feel. If it makes more sense for you to create separate mobile versions of certain sites or pages, you can use a redirect to make sure mobile device users get to the right page. To define a URL redirect, you need to specify a URL. To define a site redirect, you need only specify the site name and page name of the page to which you're redirecting. Like mobile device rules, mobile device actions are designed to be extensible. Your developers can define custom actions in addition to the four actions provided by default.

To review, if you'd like to configure an action or actions that take place when mobile device requests are received, take the following steps:

1. Create a mobile device family to represent the group of devices for which to define an action or actions.

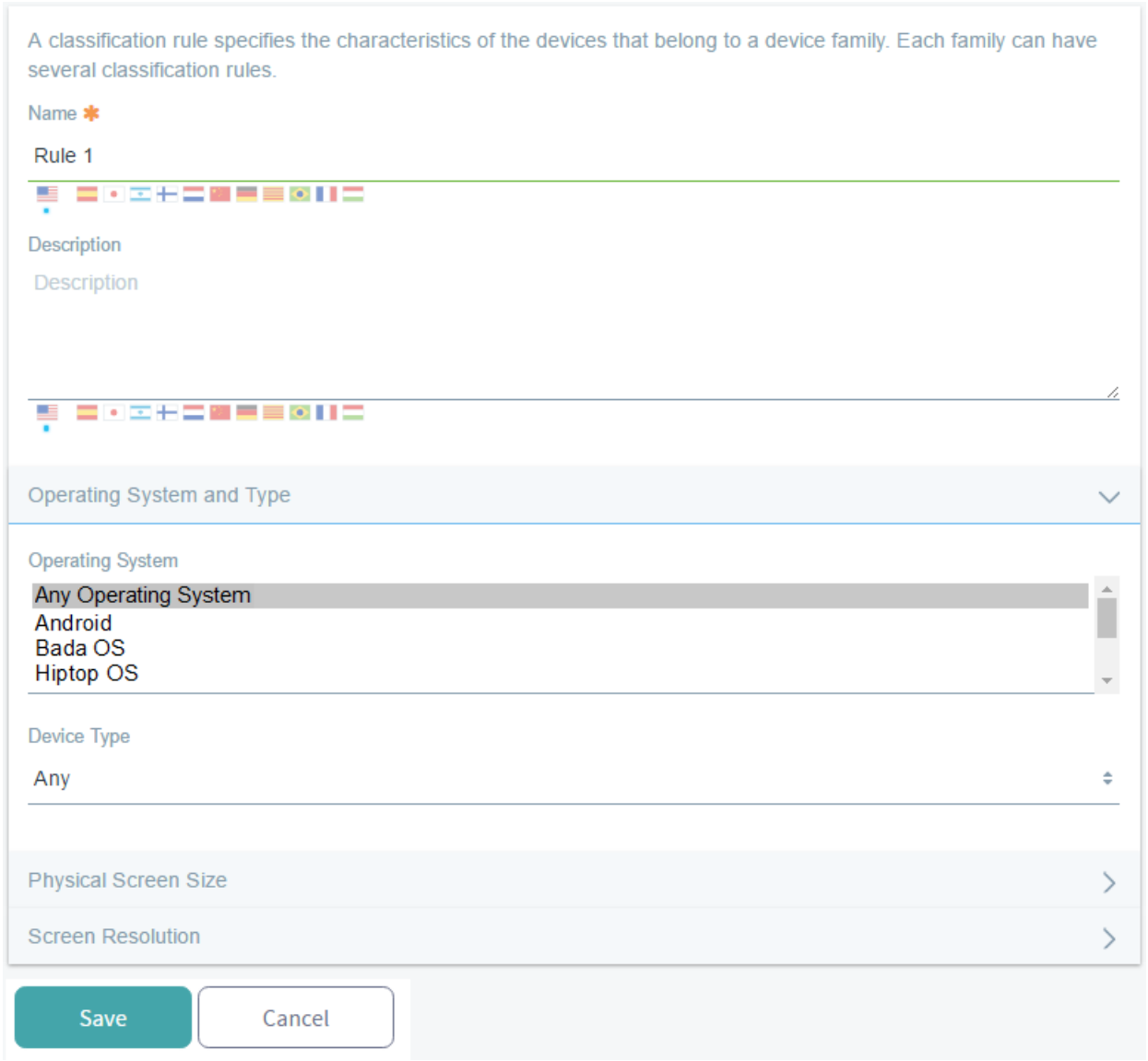


Figure 8.19: Select the operating system and device type for your rule.

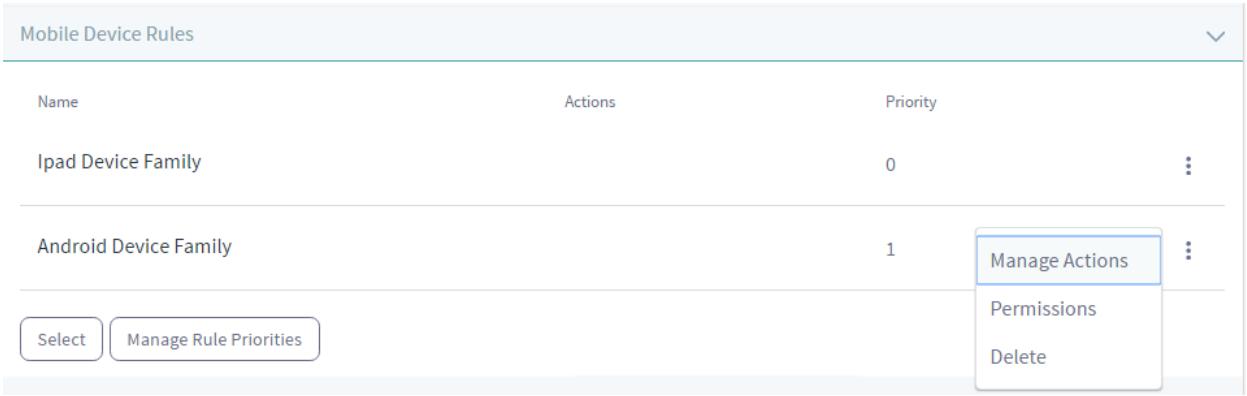


Figure 8.20: You can select a mobile device family to apply for a site or page from the Navigation section of Site Administration.

2. Define one or more rules for your family that describe the group of devices represented by your family.

3. Apply your family to an entire page set of a site (all the public pages of a site or all the private pages) or to a single page.

4. Define one or more actions for your family that describe how requests should be handled.

To see how this might work in practice, you'll study a few examples of how you can use mobile device rules. First, suppose you have a separate version of a site on your Liferay instance that's specifically designed for mobile phones running Android or Bada. For this example, you'll make a site called *Android/Bada Liferay* and you'll configure the default Liferay site to redirect incoming requests from Android or Bada mobile phones to the Android/Bada Liferay site. Yur first step is to create the Android/Bada Liferay site: go to the Sites option of the Control Panel and click *Add* () → *Blank Site*. Enter the name *Android/Bada Liferay* and click *Save*. Then add a page called *Welcome* to that site. Now your Android/Bada Liferay site has a public Welcome page just like the default Liferay site.

Next, select the default *Liferay* site in the site selector of the Menu, navigate to Site Administration → *Configuration*, and click on *Mobile Device Families*. Click on *Add Device Family* (), enter the name *Android and Bada Mobile Phones*, and click *Save*.

Click the device family link to configure the rule group to apply only to mobile phones running Android or Bada. Enter *Rule 1* for the name. Under Operating System, select *Android* and *Bada OS* (hold down Control to make multiple selections), select *Other Devices* under Tablet since you want your family to apply only to mobile phones, and click *Save*. Now you just need to define the redirect action for your family. Navigate to *Navigation*, select *Options* next to Public Pages and click on *Advanced* → *Mobile Device Rules* in the bottom navigation menu.

Click *Select* and then click the *Android and Bada Mobile Phones* device family that you configured. Once you've selected your device family, click on your device family's *Actions* → *Manage Actions*. Then click *Add Action*, enter the name *Android/Bada Liferay Redirect*, and select *Redirect to Site* under Type. Under the Site dropdown menu that appears, select *Android/Bada Liferay* and under the Page dropdown menu that appears, select the *Welcome* page that you created earlier. Lastly, click *Save*. That's it! Now Android and Bada mobile phone users are redirected to the Android/Bada Liferay site from the Liferay site.

Now you'll look at one more example of using mobile device rules before you move on. Suppose you'd like to create another rule so that when a site is accessed by an Android or iOS tablet, a different layout is used. To set this up, you need to follow the same four steps described above. First, make sure you're on the Liferay site by checking in the site selector of the Menu. Then navigate to the Mobile Device Families page of Site Administration. Add a new device family called *Android and iOS Tablets*. Add a classification rule called *Rule 1*, select *Android and iPhone OS* under the *Operating System* heading, select *Tablets* under the *Device Type* heading, then click *Save*. As with the previous example, you only need one rule to describe your device family.

Next, click on *Navigation* in Site Administration, select *Mobile Device Rules*, and select the *Android and iOS Tablets* device family. Notice that you've now selected two rule groups for the Liferay site's public pages and they've been assigned priorities. If a device making a request belongs to both of the device families represented by the rule groups, the priority of the rule groups determines which rule group's actions are executed. Note that in this example, the first rule group contains only mobile phones and the second rule group contains only tablets, so no devices can belong to both rule groups. Now you just need to define an action for your Android and iOS Tablets rule group to use a different layout: On the *Edit* page of your page group, click on *Mobile Device Rules*, and then on *Actions* → *Manage Actions* next to Android and iOS Tablets. Click on *Add Action*, enter the name *Layout Template Modification*, and select the *Layout Template Modification* action type. Lastly, select the *1 Column* layout template (or whichever one you like) and click *Save*. Good job! Now the Liferay site's pages are presented to Android and iOS tablet users with the 1 Column layout template.

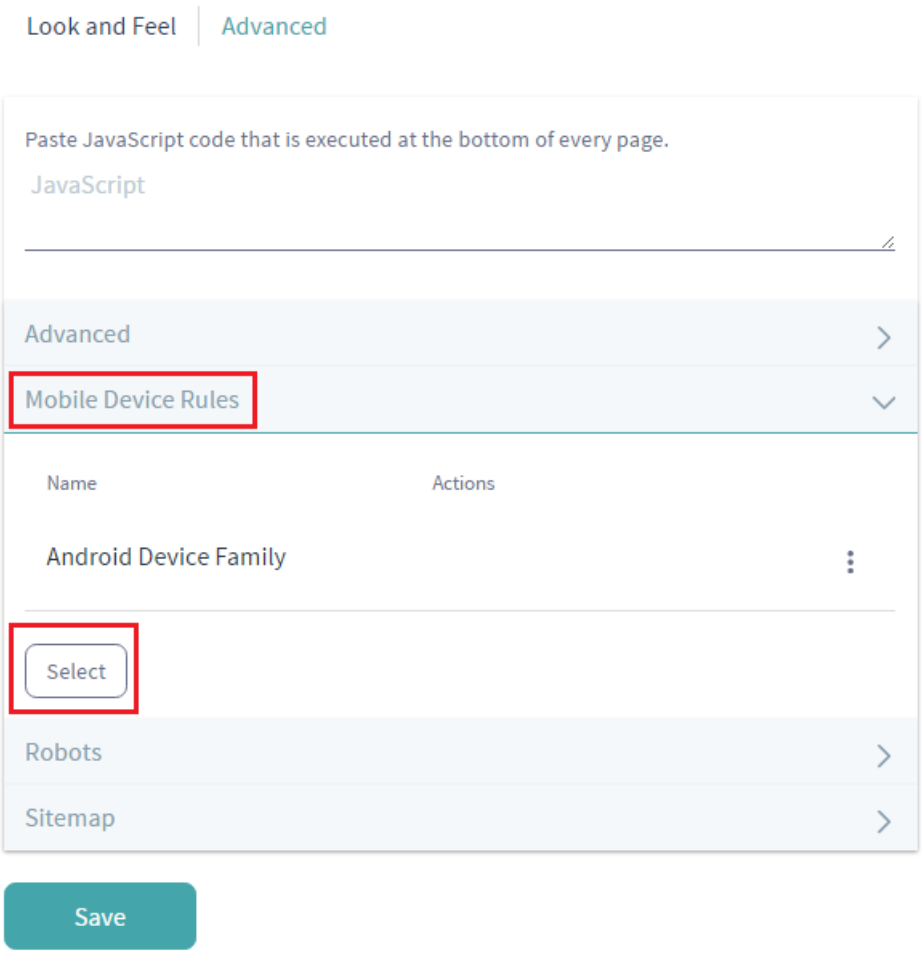


Figure 8.21: 31: To apply a mobile device family to a page set of a site, click on *Mobile Device Rules*, click *Select*, and select the desired rule group.

8.4 Building Sites from Templates

Site Templates can be administered from the Control Panel. They let Liferay administrators create multiple sites with the same default set of pages and content. Site templates can contain multiple pages, each with its own theme, layout template, applications, and app configurations. Site templates can also contain content just like actual sites. This allows administrators to use site templates to create new sites that are each created with the same default pages, applications, and content. After they've been created, these sites and their pages can be modified by site administrators. Using site templates can save site administrators a lot of work even if each site that was created from a given site template ends up being very different.

To get started, click on *Site Templates* in the Sites section of the Control Panel. Here, you can add, manage, or delete site templates. You can also configure the permissions of site templates. As long as a site is linked to the site template it was created from, changes to the site template's pages, apps, and app configurations are propagated to the site. Changes to a site template's content, however, are not propagated to existing sites that are linked to the site template. You'll learn about the propagation of changes between site templates and sites in more detail in the section on site template use cases below.

To manage the pages of a site template, click on *Site Templates* in the Control Panel and select the *Actions* icon (⋮) and then *Manage* for an existing template. If you open the main Menu on the left side of your screen (if necessary), the site template is selected in the Site Administration dropdown menu. You're provided some

similar options as a regular site which include *Navigation*, *Content*, *Configuration*, and *Publishing*. By default, the Manage Interface begins with the template's *Navigation*. From here, you can add or remove pages from a site template or select themes and layout templates to apply to the site template. Click on a specific page's *Options* icon ( ) → *Edit* from the left menu if you'd like to select a different theme or layout template for that page, or manage any of the page's extensive settings. To edit the pages themselves, click the page link from the left menu. You can add specific applications to each page of a site template and configure the preferences of each app. Each page can have any theme, any layout template, and any number of applications, just like a page of a regular site. As with site pages, you can organize the pages of a site template into hierarchies. When you create a site using a site template, the configuration of pages and apps is copied from the template to the site. By default, all changes made to the site template are automatically copied to sites based on that template.

Tip: If you want to publish a piece of web content to many sites and ensure modifications are applied to all, don't use site template content for that purpose. Instead, place the content in the global scope and then reference it from a *Web Content Display* application in each site.

The Content section offers separate repositories for content related apps based on your site template. For instance, by clicking *Polls* from the Content section, you can create a poll question that is only available for that specific site template. Assets created within your template's Content section can only be accessed by sites using the template.

The Configuration section includes Application Display Templates and Mobile Device configuration options for your site template. Also, nested in the Configuration section is the *Site Template Settings*. This option allows you to edit the template's name and description while also offering boolean options for activating your site template and allowing site administrators to modify pages associated with your template.

The following figure displays the form shown when editing the *Community Site* template's settings:

By default, the following site templates are provided:

- **Intranet Site:** Provides a preconfigured site for an intranet. The Home page displays the activities of the members of the site, search, a language selector, and a list of the recent content created in the intranet. It also provides two additional pages for Documents and Media and external News obtained through public feeds.

- **Community Site:** Provides a preconfigured site for building online communities. The Home page of a *community site* provides message boards, search, a display of a poll and statistics of the activity of community members. The site will also be created with a page for a wiki.

Now that you know the basics for creating and managing your site templates, you'll put your knowledge to the test by completing an example next.

Site Templates Example

Suppose you need to create the following three sites for the Lunar Resort's internal use: Engineering, Marketing, and Legal. These should be private sites that are only accessible to members of these respective departments. You could design each site separately but you can save yourself some work if you create a site template to use instead.

To create a site template, navigate to the Control Panel and click *Sites* → *Site Templates*. Then click the *Add* icon ( ) and enter a name for your template: you'll use *Department* for this example. Leave the *Active* and *Allow site administrators to modify pages associated with this site template...* boxes checked. The* Active* box must

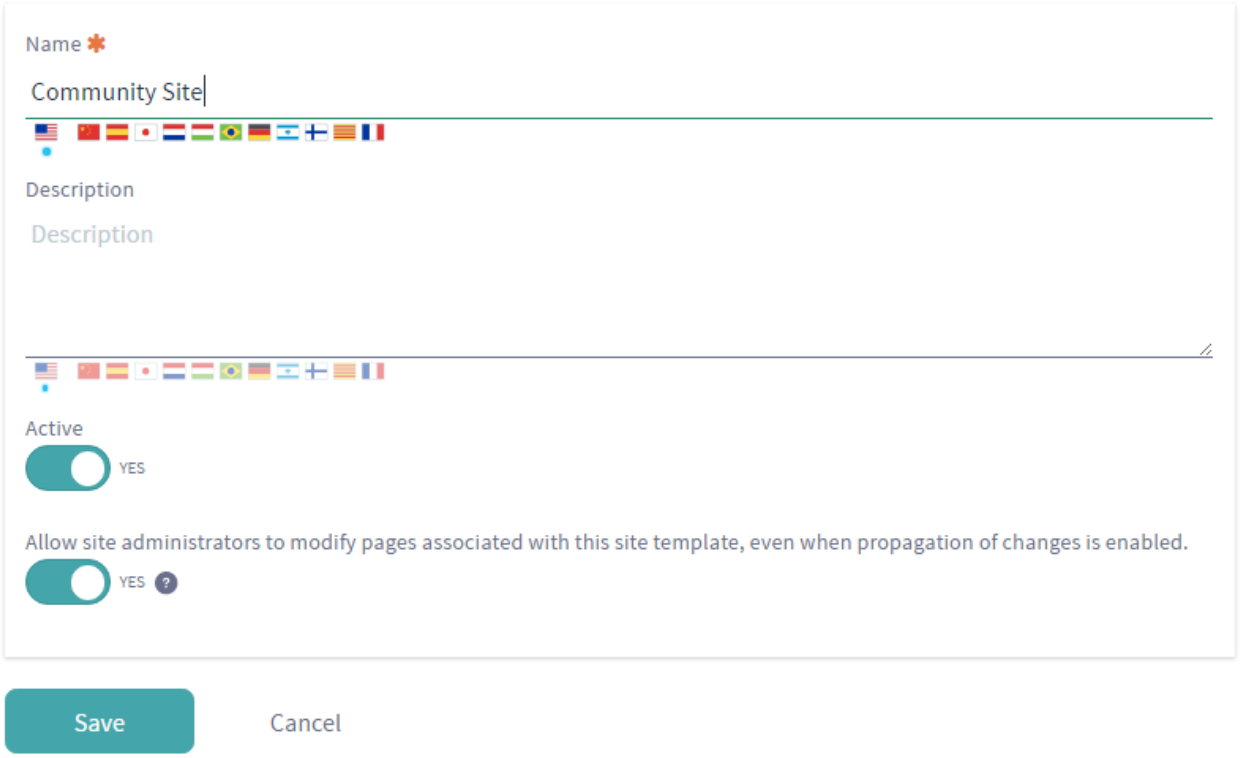


Figure 8.22: Site templates have several configurable options including the option to allow site administrators to modify pages accociated with the site template.

be checked for your template to be usable. If your template is still a work in progress, you can uncheck it to ensure that no one uses it until it's ready. Checking *Allow site administrators to modify pages associated with this site template...* allows site administrators to modify or remove the pages and apps that the template introduces to their sites–if you want the templates to be completely static, you should uncheck this. Click* Save* to create your site template.

From the left menu, notice that your site template is now selected from the Site Administration dropdown. You can now begin editing your site template. For this example, you want your site template to include four pages. First, create a *Home* page with the Activities, Announcements, and Calendar apps. Next, create a *Documents and Media* page with the Documents and Media app. Finally, create a *Wiki* page with the Wiki and Tag Cloud apps and a *Message Boards* page with the Message Boards and Tag Cloud apps. The changes you made to your site template above are completed in real time, so there's no need to navigate back to the Site Templates page of the Control Panel and select *Save*.

Next, you'll use your site template to create the Engineering, Marketing and Legal sites. Go to the Control Panel and click on *Sites* → *Sites*. Then click the *Add* icon ( ) → *Department*. Enter *Engineering* for the site name and set the Membership Type to *Private*. Recall that private sites don't appear in the My Sites application so that regular users won't even know that the Engineering site exists. Also, the only way users can be added to a private site is via an invitation from a site administrator. Leave the *Active* selector enabled so that your site can be used immediately. Select the *Copy as Private Pages* option since the Engineering site is intended for internal use only. Leave the *Enable propagation of changes from the site template* box enabled so that the Engineering site receives updates if the Department site template is modified. Finally, click *Save* to create your Engineering site.

Repeat these steps to create the Marketing and Legal sites. The new sites have all the pages and apps

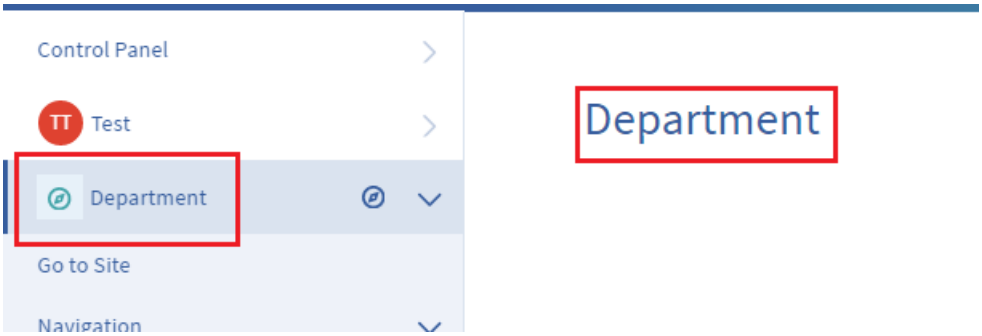


Figure 8.23: You can see the name of the site template you're currently editing.

you created in the site template. To view the pages of the new sites, click on *Sites* → *Sites* in the Control Panel and then click on *Actions* → *Go to Private Pages* next to one of your new sites. Using site templates streamlines the site creation process for administrators, making it easy to create sites quickly. Now each department of the Lunar Resort has its own calendar, documents and media library, wiki, and message boards application. Although the pages and apps of each department's site are the same, each site will quickly be filled with department-specific information as users add and share content within the sites. Also, site administrators can add new pages, apps, and content to their sites, further differentiating each department's site from the others.

Propagating Changes from Site Templates to Sites

It's possible for site template administrators to add, update, or delete site template pages. Changes made to a site template can be propagated to sites whose page sets are linked to the site template. Such a link is created when you create a site based on a site template and leave the *Enable propagation of changes from the site template* box checked. To disable or re-enable this link for a site, select the site from the Sites dropdown in the Menu by selecting the *Site Selector* button ( ). Navigate to the *Configuration* → *Site Settings* page and uncheck or recheck the *Enable propagation of changes from the site template* checkbox. In this section, you'll learn about the propagation of changes from site templates to sites and discuss the options available to site administrators and site template administrators.

If a site's page set has been created from a site template and the propagation of changes from the site template is enabled, site administrators can add new pages but cannot remove or reorder the pages imported from the site template. If a site has both pages imported from a site template and custom site pages, the site template pages always appear first; custom pages added by site administrators appear after the site template pages. Only site template administrators can remove, reorder, or add site template pages. Site administrators can add or remove custom site pages. They can also reorder custom site pages as long as they're all positioned after the site template pages. Site template administrators cannot add, remove, or reorder custom site pages.

If a site administrator changes a page that was imported from a site template and refreshes the page, the following Information icon ( ) appears in the Control Menu with the following message:

```
This page has been changed since the last update from the site template. No
further updates from the site template will be applied.
```

If the site administrator clicks the *Reset Changes* button, changes are propagated from the site template page to the corresponding site page that was imported from the site template. Clicking the *Reset Changes* button makes two kinds of updates to a page. First, changes made by site administrators to the site page

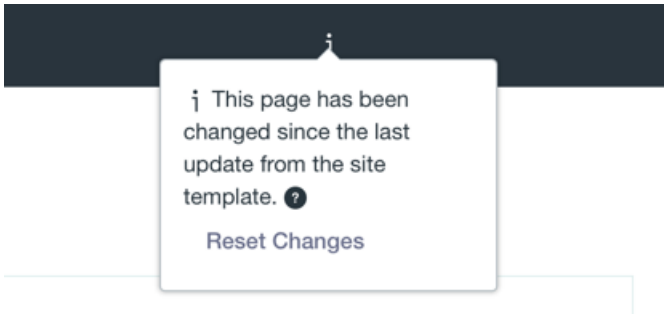


Figure 8.24: You can click the Information icon to view important information about your site template.

are undone. Second, changes made by site template administrators to the site template page are applied to the site page. Note: clicking the *Reset Changes* button only resets one page. If multiple site pages have been modified and you'd like to re-apply the site template pages to them, you'll need to click the *Reset Changes* button for each page.

Site template administrators can set preferences for apps on site template pages. When a Liferay administrator creates a site from a site template, the app preferences are copied from the site template's apps, overriding any default app preferences. When merging site template and site changes (e.g., when resetting), app preferences are copied from site template apps to site apps. Only global app preferences or local app preferences which don't refer to IDs are overwritten.

In some cases, merging site template and site changes fails. For example, if pages from a site template cannot be propagated because their friendly URLs are in conflict, Liferay DXP could try to continuously merge the site changes. Instead of entering into an infinite loop of merge fails, Liferay DXP stops the merge after several unsuccessful attempts. Liferay DXP, however, doesn't stop there: your merge is temporarily paused, you're given an indication of the current merge fail, and then you have the opportunity to fix your merge conflicts. After you've squared away your conflict, navigate to your site's *Site Administration* → *Configuration* → *Site Settings* and click the *Reset and Propagate* button.

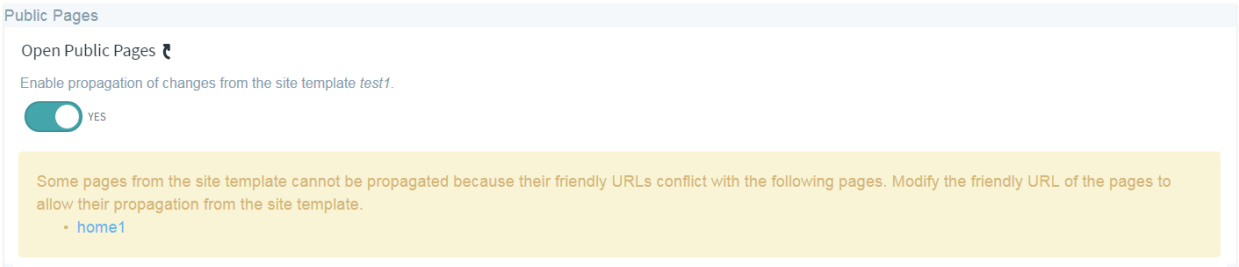


Figure 8.25: This type of warning is given when there are friendly URL conflicts with site template pages.

The *Reset and Propagate* button resets the merge fail count and attempts to propagate your site changes again. This process gives the Liferay administrator the opportunity to detect and fix a merge fail, when problems arise. This helpful process can also be done with page template merges, which follows similar steps.

Site administrators can also add data to site template applications. For example, site template administrators can add the Wiki app to a site template page and use the Wiki to create lots of articles. When a Liferay administrator creates a site from a site template, data is copied from the site template's apps to the site's apps. The preferences of the site's apps are updated with the IDs of the copied data. For example, if a

site is created from a site template that has a Wiki app with lots of wiki articles, the wiki articles are copied from the site template's scope to the site's scope and the site's Wiki app is updated with the IDs of the copied wiki articles.

Important: App data, related resources, and permissions on resources are only copied from a site template to a site when that site is *first* created based on the template. No changes made in a a template's portlet data, related resources, or permissions are propagated to the site after the site is created. Neither are such changes propagated to a site by the *Reset* or *Reset and Propagate* features.

For example, consider a site template administrator who includes a Message Boards app as part of a site template. They even create Message Board categories and configures permissions over the actions of the categories. The first time a site is created based on the site template, the categories (app data) and related permissions are copied to the site. If the site template administrator adds, removes, or deletes some categories, however, such changes *aren't* propagated to the site.

Now that you've learned how site templates work, you'll learn how to share site templates.

Sharing Site Templates

If you want to export a site that uses site or page templates to a different environment (through a LAR file or remote publication), the templates must be exported and imported manually in advance or the import will fail.

To export a Site using a Site Template, use the following process:

1. Go to the *Control Panel* → *Sites* → *Site Templates* menu.

2. Click the *Actions* icon (⋮) and then *Export* for the site template your site is using. You'll use the *Export* screen to obtain a LAR file with the content of the site template. Be sure to choose the applications and data you want exported.

3. In your target environment, go to *Control Panel* → *Sites* → *Site Templates* and create a new site template.

4. Click *Actions* → *Import* for that site template and upload the LAR file containing your site template's content.

Now the site can be exported and imported normally to this new environment. For more information on exporting/importing content, visit the Importing/Exporting Pages and Content article.

In the next section, you'll learn about page templates.

8.5 Creating Pages from Templates

Page templates function similarly to site templates but at the page level. Each page template provides a pre-configured page to reuse. Within a page template, it's possible to select a theme or layout template, and add applications to the page and configure app preferences. Both sites and site templates can utilize page templates for creating new pages. Click on *Sites* → *Page Templates* in the Control Panel to see a list of page templates.

You can edit or delete existing page templates, configure their permissions, or add new page templates. By default, three sample page templates are provided:.

- **Content Display Page:** provides a page preconfigured to display content. It has three auxiliary applications (Tags Navigation, Categories Navigation, and Search) and an Asset Publisher. The most significant aspect of this page is that the Asset Publisher is preconfigured to display any web content associated with this page. This means that you can select any page created from this page template as a *Display Page* for a web content article. You can choose a display page for a web content article when creating a new web content article or when editing an existing one. When you create a new web content article, a unique (canonical) URL for the web content pointing to this page will be assigned to it.
- **Wiki:** provides a page with three applications related to authoring a wiki. It also has two columns, the main left column with the Wiki application and two right side appss to allow navigating through pages by tags and categories.
- **Blog:** provides a page with three applications related to blogging. It has two columns, the main left column contains the Blogs application and the small right column provides two side apps, Tag Cloud and Recent Bloggers. The Tag Cloud application shows the tags used within the site and allows navigating through the blog entries shown in the main Blogs app.

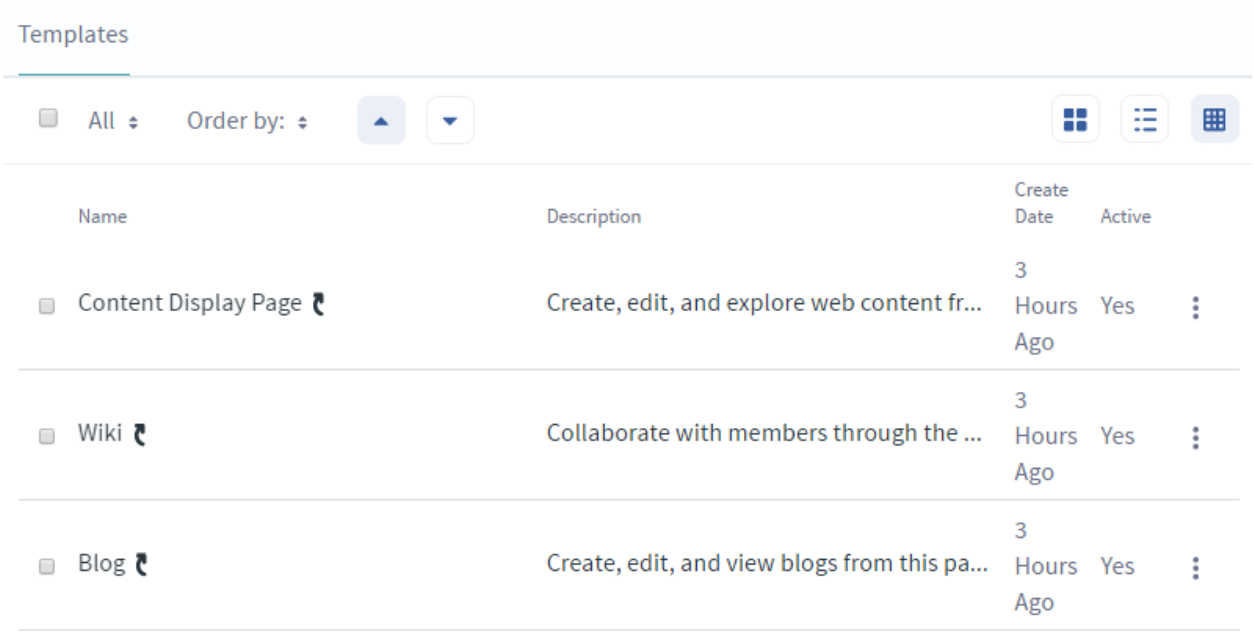


Figure 8.26: The Blog page template is already available for use along with the Content Display Page and Wiki page templates.

To add a new page template, click the *Add* icon ( ). Then enter a name and description for your template. Leave the *Active* button enabled. Click *Save* and then identify your page template in the list. Use the *Actions* icon ( ) to edit the page template. Clicking the page template's name opens a new browser window which you can use to configure your new page. Any changes you make are automatically saved so you can close the new browser window once you're done.

Note that after a new page template has been created, the default permissions are to only allow the creator to use the page template. To give other users access to it, use the Actions menu in the list of templates and choose *Permissions*. Once you see the matrix of roles and permissions, check the *View* permission for

the role or roles needed to see the page template in the list of available page templates when creating a new page. If you want any user who can create a page to be able to use the page template, just check the *View* permission for the *User* role.

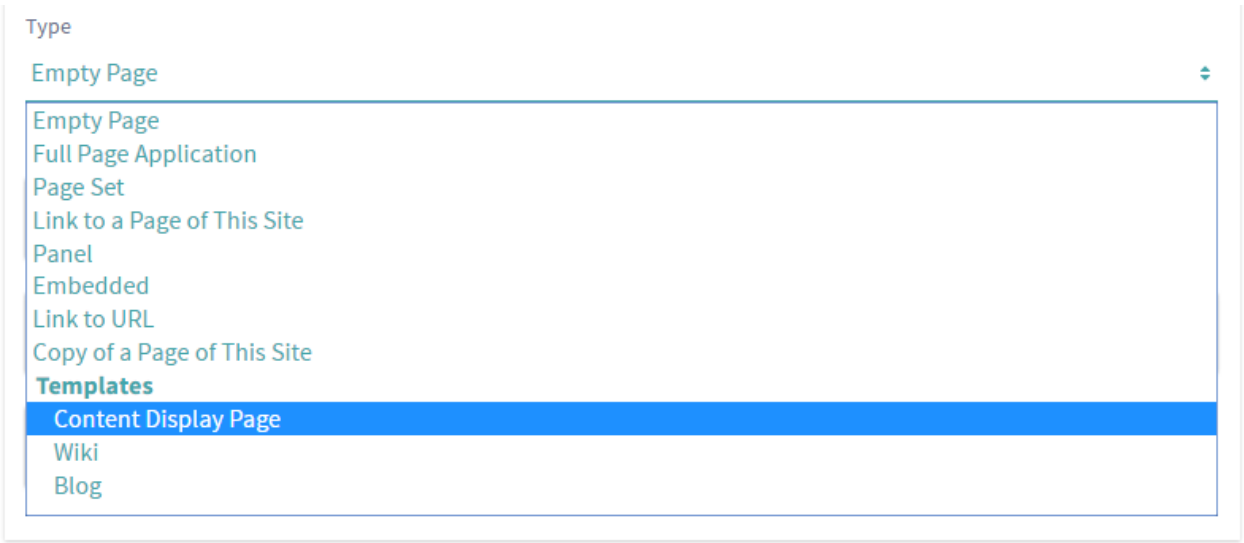


Figure 8.27: When creating a new site page, you're given options for the page template and page type.

To use your template to create a new page, just navigate to your site's Site Administration dropdown menu and select the *Navigation* dropdown option. Select the *Actions* button (⋮) for the page or page set you'd like to add a page to and then click the Add Page button. You'll be able to select a page template and type a name for the new page.

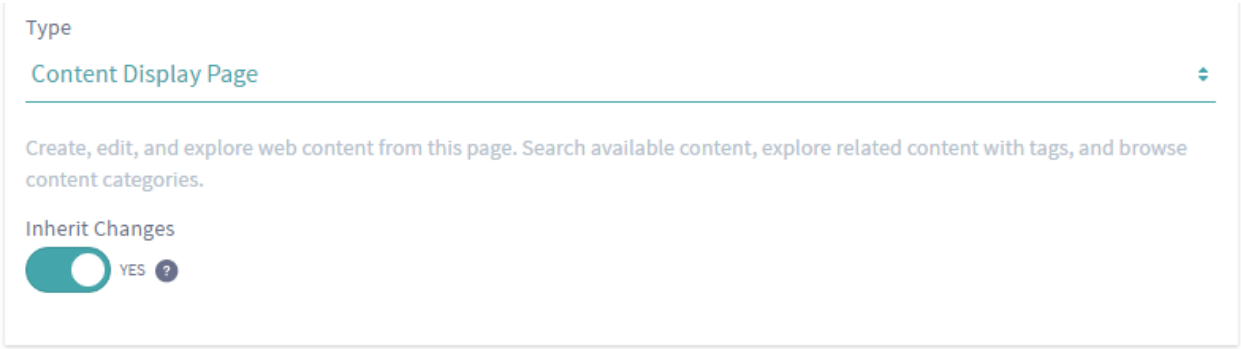


Figure 8.28: You can choose whether or not to inherit changes made to the page template.

Note that by default, when a site administrator creates pages based on a page template, any future changes to the template are automatically propagated to those pages. Site administrators can disable this behavior by disabling the *Inherit Changes* selector. Occasionally, propagation for page templates fails due to unintended errors. To learn how to manage a failed page template propagation, visit the Propagating Changes from Site Templates to Sites section of this chapter.

If staging has been enabled, changes to the page template are automatically propagated to the staged page. These changes still need to be approved before the page is published to live. For this reason, the

automatic propagation of page template changes to the staged page cannot be turned off and the *Inherit Changes* selector does not appear.

You'll learn about staging later in the User's Guide. For now you'll look at importing and exporting templates.

Sharing Page Templates

If you want to export a page that uses a page template to a different environment (through a LAR file or remote publication), the template must be exported and imported manually in advance or the import will fail.

To export a page using a page template, use the following process:

1. Go to *Control Panel* → *Sites* → *Page Templates*.
2. Next to the page template you would like to export, click *Actions* → *Export*. This produces a LAR file you can import later.
3. On the target environment, go to *Control Panel* → *Sites* → *Page Templates* and create a new page template.
4. Next to the new template, click *Actions* (⋮) → *Import*.
5. Upload the LAR file containing the exported page template from step 3.

The page template can now be imported normally to your new environment. For more information on exporting/importing content, visit the Importing/Exporting Pages and Content article.

Next, you'll examine the tools Liferay DXP provides for exporting/importing content.

8.6 Importing/Exporting Pages and Content

Liferay DXP's Export/Import feature gives you the power to backup and restore your site and app data. The export feature grants users the flexibility of exporting site or app-specific content they've created as a LAR (Liferay Archive) file to other Liferay instances, or to save it for a later use. The import feature can be used to ingest the LAR file you exported from Liferay, which restores the work you previously exported.

For example, suppose you're managing a site that celebrates the Thanksgiving holiday every year by creating a *Thanksgiving* themed page during the month of November. Every November, you'd like to publish the holiday themed page, but want it removed after Thanksgiving ends. Instead of manually creating the page every year, just to delete it and create it again next year, you can use the Export/Import features to streamline the process. When the holiday ends every year, you can export the page as a LAR file and save it outside your site. Then when the holiday season approaches the following year, you can import the page, make some minor tweaks, and publish it with little effort.

There are two primary places Export/Import is used: sites and apps. You can learn more about exporting/importing app data in the Exporting/Importing App Content section. In this section, you'll learn how to export and import content for sites.

Backing Up and Restoring Pages and Their Content

In the Site Administration dropdown of the Menu, there is an option called *Publishing Tools*, which is where the *Export* and *Import* features reside for pages. If you click on *Export*, you are presented a simple interface which can be used for exporting your public or private pages. The Export feature allows you to export your site's data as a single LAR file. Similarly, if you click *Import*, you're provided a similar interface which can be used for importing public or private pages as a LAR file.

When importing data into a site, it's best to use a newly created site to avoid potential conflicts between the existing site data and the data about to be imported. When exporting site data, you can specify exactly what data should be included in the LAR:

- Site pages (you can select exactly which ones)
- Page settings
- Theme
- Theme settings
- Logo
- Application configurations
- Application content
- Archived setups
- User preferences

Once you've created a LAR file, you can import it into a site on another Liferay server. The data included in the LAR file, including all the site pages, will be imported into the site. Exporting and importing LARs is a great way to take content from a site in one environment (say, a development or QA environment) and move it all in one shot to a site on another server. You can use LARs to import data onto production servers, but you should not make this a regular occurrence. If you want to regularly move pages from one server to another, you should use Liferay DXP's staging environment, which is discussed in the Staging Content for Publication section.

LARs can be a good way to back up your site's content. You can export them to a specific location on your server which is backed up. If you ever have to restore your site, all you need to do is import the latest LAR file. However, please be careful! If there's any content that exists both in the LAR and in the site that's importing the data, there may be a conflict, and data could be corrupted. If you'd like to restore a Liferay site using a LAR file, it's best to delete the site entirely, create a new site with the same name as the old one (i.e., re-create the site), and then import the LAR file into the new site. This way, there's no chance for there to be a data conflict.

Liferay DXP can handle some kinds of naming collisions when importing a LAR file into a site. For example, suppose you're importing a LAR file into a site and the LAR file has a page with a certain friendly URL. If an existing page in the site has the same friendly URL there will be a collision. Liferay DXP resolves the collision by adding a number to the end of the friendly URL and incrementing until there's no collision. This behavior takes place for friendly URL translations as well. Similarly, if importing a LAR into a site causes a category name collision, Liferay DXP renames the imported categories.

Note: LAR files are version dependent. You can't import a LAR file that was exported from one version of Liferay into a Liferay server that's running a different version of Liferay. Also, note that periodically exporting LARs is *not* a complete backup solution; please refer to the Backing up a Liferay Installation section for information on backing up Liferay.

Next, you'll simulate being a good administrator and exporting a LAR file for backup purposes. Click on the *Export* button from the *Publishing* menu and click the *Add* button ( ). A *New Export* page loads, allowing you to configure what pages and content you'd like to export from your site.

Give your export process the name *Lunar Resort Version 1*. Use the *Pages* category to select public or private pages and their settings you'd like to export. Also notice the *Content* category, which lets you choose all content or specific content for your selected pages.

For this initial export, select everything. Note that if you select one of the *Choose* radio selectors or *Change* links, you're given checkboxes for options to choose. The applications' content can also be selected for export,

including the Documents and Media Library, Message Boards, and Web Content assets. You can even export the theme you're using! Lastly, you can select whether the permissions for your exported pages and content are included.

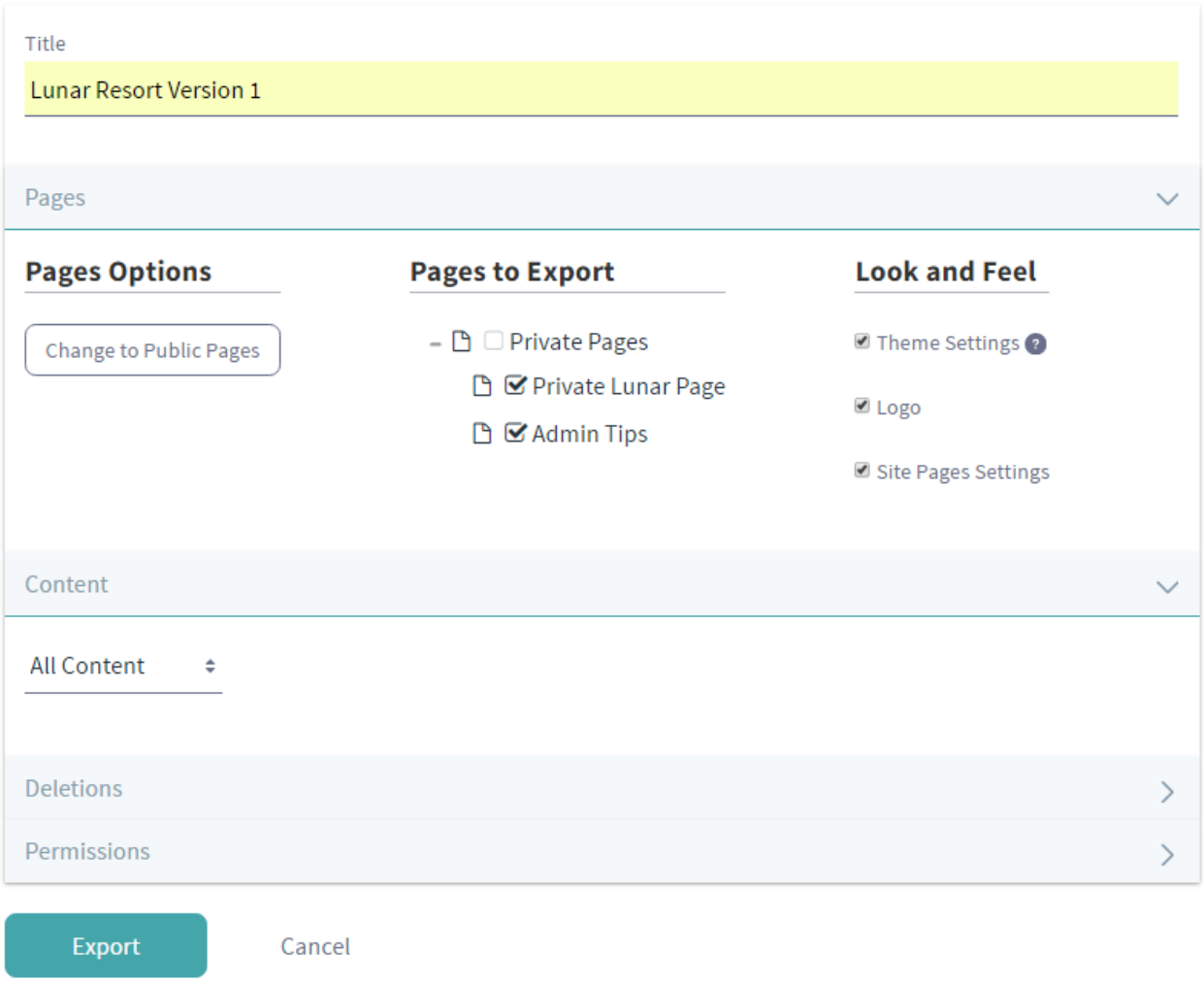


Figure 8.29: You can configure your export options manually by selecting pages, content, and permissions.

Once you click *Export*, the menu automatically switches to the *Processes* tab, where you'll see the status of your exported LAR file. You can select the *Download* icon ( ) to download the export to your local machine. Once you have the file, you can copy it to a backup location for safekeeping or import it into another installation of Liferay. If you must rebuild or wish to revert back to this version of your site, you can import this file by clicking the *Import* button from the Publishing menu, browsing to it, and selecting it. You also have the option of dragging a LAR file inside the dotted area, which also executes the import process.

Another useful option to use when exporting content is the *Export Templates* feature. The exercise you completed previously created a custom export process. Instead of manually having to customize an export process every time you're looking to export pages/content, you can use an export template. Using export templates provides you the convenience of storing export process settings so they can be reused. If you export pages frequently and usually select the same options to export, the task of selecting options repeatedly can become tedious. With export templates, you can select a custom template and immediately export with

the options you configured.

To create an export template, select the *Options* icon ( ) from the top right corner of the screen and select *Export Templates*. Click the *Add* button ( ) and assign the template a name and description, and then fill out the configuration options as you would during a custom export process. Once you've saved your export template, it is available to use from the *Export Templates* menu. To use the template, click the *Actions* button ( ) next to the template and select *Export*. This automatically fills the fields and options for exporting pages and their content. All you have to do is give the export process a custom name. Once you click *Export* to confirm the configuration settings, your LAR file is generated.

Next, you'll discuss creating site teams.

8.7 Creating Teams for Advanced Site Membership Management

If you have an ad hoc group of users who perform the same set of tasks in a site, you can organize them into Site Teams. Site administrators can assign these team's permissions for various site-specific functions. Site Teams are the preferred method for collecting permissions within a single site. Some common functions to assign a Site Team include:

- Moderating site Wiki content
- Managing Message Boards threads
- Writing blogs
- Editing a specific page in the site

For instance, if your site has Message Boards, you might want to enable a subset of the site's members to moderate the categories and threads, and perhaps to ban abusive/offensive posters. To do this, you could create a Site Team named *Lunar Resort Message Board Moderators*, define the team's permissions in the Message Boards application, and assign the desired site members to the team.

The permissions assigned to a Site Team only apply to that site. Knowing that a team's permissions don't impact other sites, site administrators can concentrate on defining and applying permissions to their sites' teams.

Note: To create and apply permissions for a group of users to use across multiple sites or organizations in your Liferay instance, consider aggregating the users into a User Group and assigning the User Group permissions via Roles.

To create a team within a site, first navigate to the Site Administration page of your site and select *Members → Site Teams*. It's important to note that configuring other site membership groupings, such as *Users*, *Organizations*, and *User Groups* can be done in the *Site Memberships* app, which is also located in the Members tab. You can visit the User Management chapter for more information on how these site memberships work. Finally, click the *Add Team* icon ( ).

After you've clicked the *Add Team* button and entered a name and a description, click *Save*. Your new team shows in the list. To add members, click on the team name link and then select *Add Team Members*.

To manage a team's permissions, click on the *Actions* icon ( ) and select *Permissions* for that team. Setting permissions for the team assigns all of the team's members those permissions. Only administrators with the ability to edit/manage the team have the ability to manage team permissions.

If you created a team whose task is to moderate the Message Boards, for example, you'd want to give the team all the permissions they'd need. To do this, navigate to *Site Administration → Content → Message Boards*

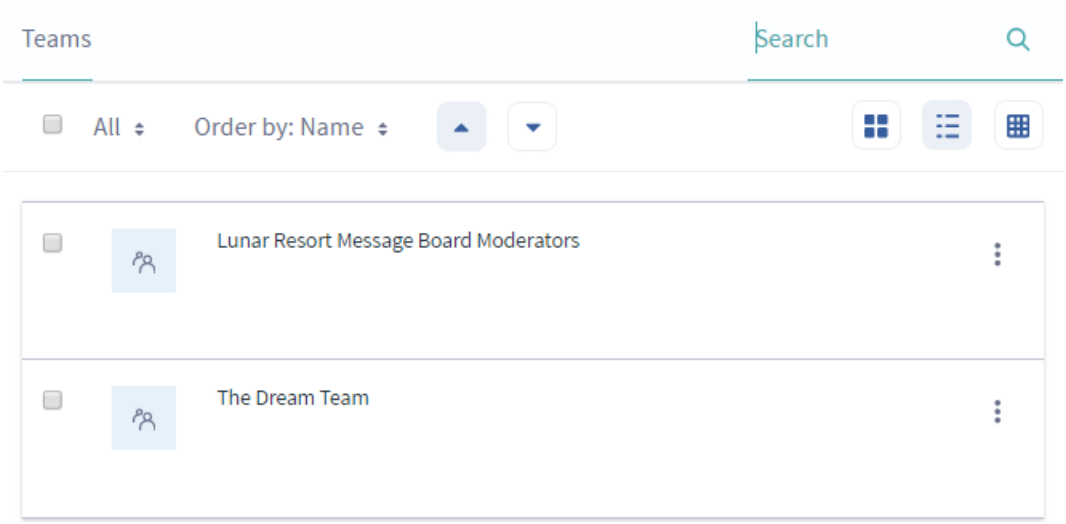


Figure 8.30: Creating teams within your site can foster teamwork and collaboration, as team permissions enable team members to access the same resources and perform the same types of tasks.

and select *Permissions* from the *Options* icon ( ) in the top right of the screen. Find the team in the Role column, and select the appropriate permissions.

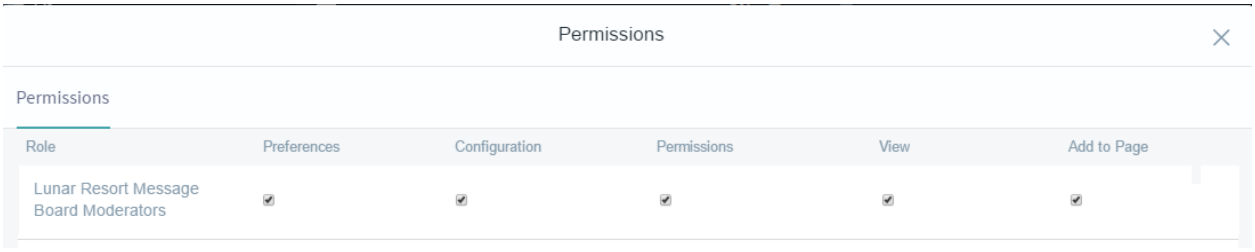


Figure 8.31: The Lunar Resort Message Board Moderators Site Team has unlimited permissions on the Message Boards application.

That's it! It's easy to give groups of site users appropriate permissions to perform their tasks. This chapter has provided an introduction to Liferay DXP site management. You've learned how to use Liferay DXP to create multiple sites with different membership types. You've also seen how easy it is to create and manage sites and to create and manage pages within a site in Liferay DXP. Next, you'll begin working with web content.

Chapter 9

Creating Web Content

Liferay DXP's Web Content Management (WCM) system allows non-technical users to publish content to the web without having advanced knowledge of web technology or programming of any sort. Liferay WCM empowers you to publish your content with a simple point and click interface and it helps you keep your site fresh. You'll find yourself easily creating, editing, and publishing content within just a few minutes of being exposed to its features. Liferay WCM, however, doesn't sacrifice power for simplicity. If need be, you can use your developer skills to create complex presentation layer templates that make your content "pop" with dynamic elements. Once these templates have been deployed into Liferay DXP, your non-technical users can manage content using these templates as easily as they would manage static content. All of this makes Liferay WCM an appropriate choice both for sites with only a few pages and for sites with gigabytes of content.

Nearly all Liferay users use Liferay DXP's Web Content Management system. After all, every web site has content that needs to be managed. Liferay DXP's WCM empowers you to manage all the content on your site quickly and easily within your browser. Beyond managing existing content, Liferay DXP's WCM lets users easily create and manage everything from simple articles containing text and images to fully functional web sites. Web publishing works alongside Liferay DXP's larger collection of applications, which means you can add shopping cart functionality, visitor polls, web forms, site collaboration tools, and more. Everything is done with our collection of easy-to-use tools with familiar rich-text editors and an intuitive interface.

Liferay DXP's WCM offers a host of features that makes managing the content of your site easier. A list is provided below of some of the tools that can be used with web content in Liferay. Some of these tools are covered in other chapters of the User's Guide, so links are provided if you're interested in using them to manage your site's web content.

- **WYSIWYG Editor:** A complete HTML editor that allow you to modify fonts, add color, insert images, and much more.
- **Structure Editor:** Easily add and remove fields you want available to content creators and then dynamically move them around. This editor includes an entire suite of form controls you can drag and drop onto your structure.
- **Template Editor:** Import template script files or create your own template that informs the system how to display the content within the fields determined by the structure.
- **Web Content Display:** An application that lets you place web content on a page in your site.

- **Asset Publisher:** An application which can aggregate different types of content together in one view. This app is covered in more detail in the Publishing Assets section.
- **Scheduler:** Lets you schedule when content is reviewed, displayed and removed. This feature is covered in more detail in the Scheduling Web Content Publication section.
- **Workflow Integration:** Run your content through an approval or review process. This feature is covered in more detail in the Using Workflow section.
- **Staging:** Use a separate staging server or stage your content locally so you can keep your changes separate from the live site. This feature is covered in more detail in the Using the Staging Environment section.

With Liferay DXP's WCM, you have the ability to create, edit, stage, approve, and publish content with easy-to-learn yet powerful tools. Liferay's WCM streamlines the content creation process for end users. It's much faster to use Liferay's WCM than it would be to create all the content for your site in HTML. WCM is integrated with Liferay's services so advanced template developers can use them to query for data stored elsewhere in Liferay.

By the time you're done, you'll be able to apply all these concepts to your own content. To demonstrate Liferay DXP's Web Content Management features, you'll create and manage content on Liferay for the ambitious (and fictitious) *Lunar Resort* project. The Lunar Resort project specializes in facilitating lunar vacations. It provides space shuttle transportation from the Earth to the Moon and back, offers the use of a state-of-the-art recreational facility enclosed by a large, transparent habitat dome, and even rents out lunar rovers. Once you're familiar with Liferay WCM, you'll wonder how the Lunar Resort instance could ever manage without it!

9.1 Publishing Basic Web Content

Liferay DXP's Web Content Management is a powerful and robust tool for creating and organizing content on your web site. You'll begin by examining some basic concepts involving sites and pages.

As you'll see, Liferay DXP's WCM is a full-featured solution for managing your web site. You'll start with an overview of what it has to offer and then you'll dive down into its features. Note that web content is just one kind of asset on Liferay. Other types of content (blog posts, wiki articles, message board posts, etc.) are also considered assets. Liferay DXP provides a general framework for handling assets that includes tags, categories, comments, ratings, and more. Please see the Publishing Content Dynamically section for more information on Liferay's asset framework.

Creating Web Content

Content is the reason web sites exist. Liferay DXP has made it easier than ever to get content published to your site. Because Liferay DXP is so flexible, you can use basic authoring tools right away or take advantage of the more advanced features. It's adaptable to your needs.

You'll begin by creating some simple content using Liferay's WYSIWYG editor. Then you'll publish it to the home page of the Lunar Resort's web site. This is a fast and straightforward process that demonstrates how easy it is to create and publish content on your Liferay instance. You'll learn about the Web Content section in Site Administration so you can create and publish your first pieces of content.

When you manage web content from the Site Administration menu, you can select the location where the content resides. When selecting the Site Administration dropdown from the Menu, you are presented with two scopes: site scope and page scope. The site scope can be managed by clicking the *Site Selector* button

( ) located on the Site Administration dropdown menu, which is characterized by the name of the site. From there you can select the site for which you want your content scoped. For instance, you can add content that's available to a specific site or globally across your Liferay instance. By default, the page scope must be configured before you can access it. For instance, if you add a Web Content Display app to a site page called *Lunar Rover*, you can navigate to the app's *Options* icon ( ) and select *Configuration* → *Scope*. From the scope dropdown, you can select the current page you're on, which will be characterized as *New* in parenthesis. Click *Save* and return to the Site Administration → *Content* menu. You'll now observe the *Default Scope* option. Select the *Default Scope* icon ( ) and choose the page you configured. Now the content created in this app is scoped to the *Lunar Rover* page only. For more information on scoping content in an application, visit the Application Scope section.

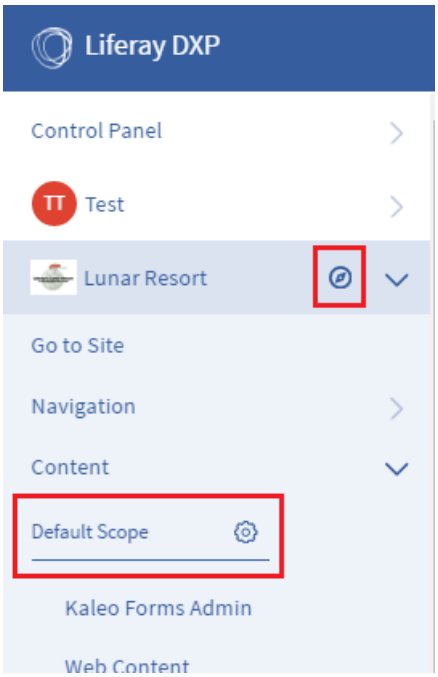


Figure 9.1: You can choose where to create content by navigating to the Site Administration menu and selecting your site and page scope.

Once you have the Lunar Resort site selected, click on the *Web Content* link under Content. You'll see a folder structure containing all of the web content articles that exist in the currently selected scope (the Lunar Resort site). You can click the *Add* icon ( ) → *Folder* to create a new folder. For sites with lots of content and web content articles, it can be very useful to use folders to group certain kinds of web content articles together. Click *Add* → *Basic Web Content* to create a new web content article.

Figure 9.2: Click *Add* → *Basic Web Content* to create a new simple web content article. To create a new web content article based on an existing web content structure, click *Add* and then click on the name of the structure you'd like to use.

Existing web content structures also appear in the *Add* menu. This provides users with shortcuts for creating specific kinds of web content articles. For example, if a web content structure called *FAQ* has been

created for Frequently Asked Questions articles in your currently selected scope, you can create a new FAQ article by clicking *Add → FAQ*.

Note: In previous versions of Liferay, you could specify web content types via `portal.properties`. In Liferay DXP 7.0, web content types are no longer used and have been replaced by vocabularies. Vocabularies allow users to filter their web content articles by category instead, which lets you filter your content using the Asset Publisher and faceted search. To learn more about vocabularies and how to use them with web content articles, see the Defining Categories for Content section.

You can provide a structure and template to your web content articles. You'll learn more about the power of web content structures and templates later. For now, you'll cover the basics of creating a piece of web content by first exploring the editor.

Using the WYSIWYG Editor Once you've clicked *Add → Basic Web Content*, you'll find a highly customizable form that, by default, has three fields: title, summary, and a powerful WYSIWYG editor. You're also provided a boolean *Searchable* switch. You could customize this form to contain whatever fields your content needs but you'll keep things simple for now. If web content structures have already been created in your currently selected scope, you can select one for your new web content article by clicking the *Structure and Template* dropdown. You'll discuss web content structures and templates in detail in the next chapter.

Getting a new web site up and running is an exciting step for anyone, whether it is a large corporation or a small non-profit charity. To celebrate this momentous achievement at the Lunar Resort, you'll give our announcement some of the pomp and circumstance it deserves!

Type the words *Welcome to the Lunar Resort* in the *Title* field. In the *Summary* field, give a short description of the Lunar Resort's facilities. In the *Content* field, you'll add the body of your web content article, which you'll dive into next. Lastly, leave the *Searchable* switch enabled.

Note: Disabling the *Searchable* switch for an article prevents it from being indexed. This prevents it from appearing in search results or in the Asset Publisher. The article is visible, however, to an Administrator in the list of web content articles displayed in Site Administration or the Web Content Display portlet.

The kneejerk reaction to the simplistic looking WYSIWYG editor is "Where are the editor's controls?" Don't let the simplistic look of the editor fool you; the editor gives you a seamless writing experience, displaying controls when you need them and hiding them from view when they're unnecessary. This keeps the editor space uncluttered so you can focus on your main objective: writing. As you create content, the context-specific controls appear.

First, add some text that will serve as the heading in your article. If you highlight the text, controls appear. These controls let you style the text, provide a link, or share the article on Twitter. For your heading text, select the *Styles* dropdown and give your heading a *Heading 1* style.

Whenever you place your cursor in the content area, the *Add* icon ( ) appears. If you click on it, controls for inserting an image, table, or horizontal line ( ) appear. To insert an image, select the icon that depicts a mountain silhouette. The image file selector screen appears, allowing you to choose an existing image or upload a new one. If you select an existing image in your Documents and Media repository, you can access the image editor, via the pencil icon ( ) in the bottom right corner of the preview window, to make changes to the image. Once you've made edits, a copy of the image is automatically created for you to use in your web content.

After adding an image to the web content article, clicking it brings up controls ( ) for justifying it to the left, center, or right side of the article. You can also make it a link and define the `alt` HTML attribute.

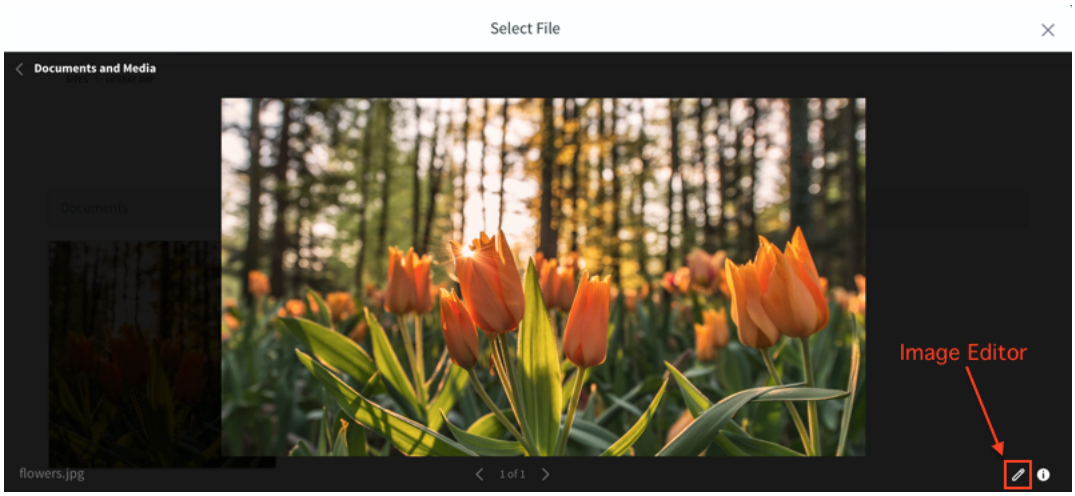


Figure 9.3: You can access the image editor through the item selector window.

Go ahead an add an image to the Lunar Resort article, to spice it up a bit. Note that when adding an image via the web content editor, you can either select the file from the Documents and Media app or provide it via a URL.

You can also insert a table with as many rows and columns as you see fit. When clicking inside the table, table editing controls appear. They let you designate the first row and/or column as table headers, and also enable you to add rows, columns, and cells. You also have the option to insert a horizontal line, which is a good seperator between sub-articles or an article and its title. Now you're familiar with the editor's regular mode.

For those content creators that would rather write in HTML code, the editor also caters to those individuals. To switch the editor to source view, select the *Source* icon ( ). Note that the regular mode icon ( ) appears, which you can select to return to regular mode. You also have the option to switch between a dark and light theme by choosing the moon and sun icons. The built-in syntax coloring helps you identify HTML elements, regardless of the mode you're using.

You can even work in a dual pane view that shows your HTML code on the left and a preview pane on the right. To open this view, click on the *Enlarge* icon ( ). You can arrange the HTML and preview panes horizontally or vertically. You can also hide the preview pane, if preferred. You can exit the enlarged editor by clicking the *Done* button at the bottom of the screen.

Add a few short sentences announcing the grand opening of the Lunar Resort. The content can be localized in whatever language you want. You'll learn more about localizing your content later on.

You can integrate Liferay DXP with external services to enable additional functionality. For example, if you navigate to the Control Panel, click on *Configuration → Server Administration → External Services*. From this menu, you can install and enable Xuggler. Enabling Xuggler allows you to embed audio and video files in web content. Installing and enabling Xuggler is easy; you can do it right from the Control Panel. Please refer to the Publishing Files article of this guide for more details.

Once Xuggler has been installed and enabled, embedding audio or video files in a web content article is easy. By default the current WYSIWYG editor (AlloyEditor) does not provide audio/video files. You can extend the default AlloyEditor by adding an audio/video button. You can learn about doing this in the WYSIWYG Editors tutorials section. Another option you have is changing the WYSIWYG editor to one that supports embedding audio/video files in web content. The CKEditor, for example, is an editor that provides this functionality. To use the CKEditor, create a `portal-ext.properties` file in your Liferay DXP root folder and add the following property:

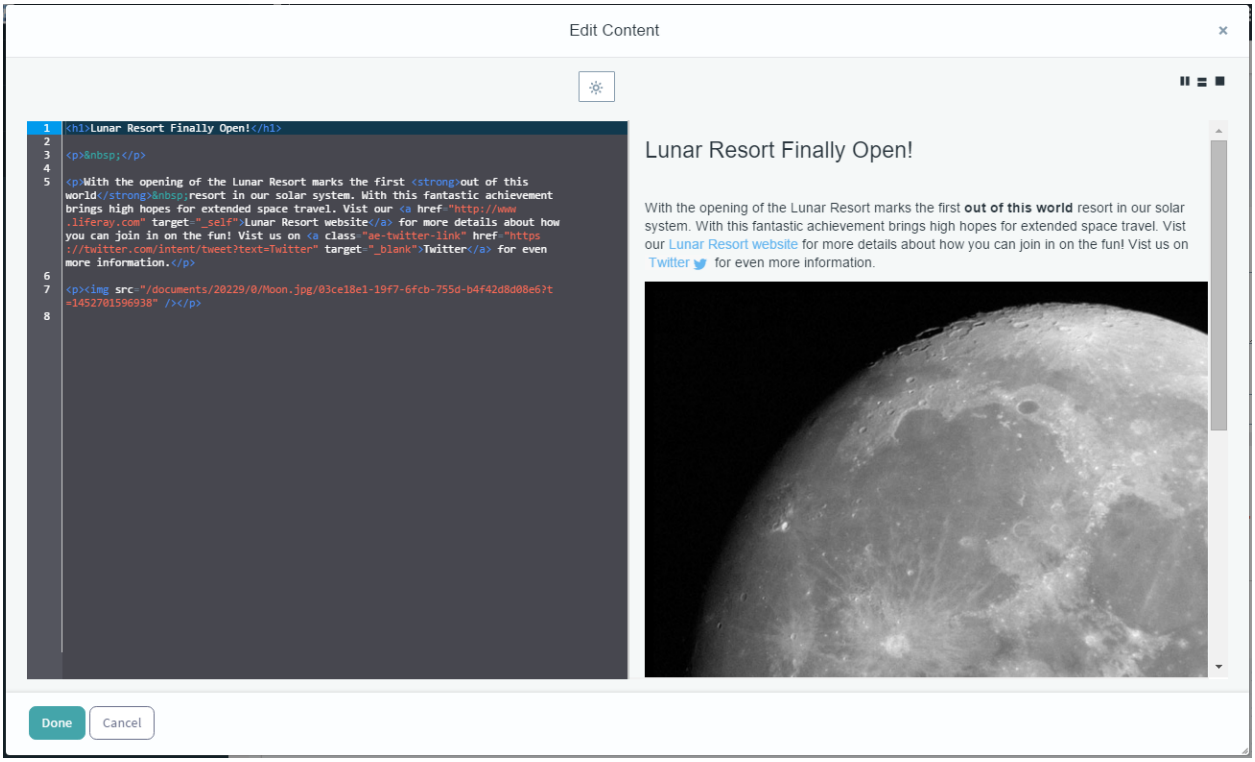


Figure 9.4: You can view how your HTML would render by using the preview pane.

```
editor.wysiwyg.portal-impl.portlet.ddm.text_html.ftl=ckeditor
```

Once you restart your Liferay instance, the AlloyEditor is replaced with the CKEditor and you have the ability to add audio/video files!

Place your cursor in the editor and select the audio/video button and then choose the file you'd like to insert. If you haven't already uploaded the audio or video file to your Liferay instance, you'll need to navigate to *Documents and Media* in the Content section and upload the file to Liferay. Select the file and then check that the audio or video component appears in the web content. Excellent! When your web content is published, users can view or listen to the embedded multimedia!

You can also download the web content article in XML format by clicking the *Options* icon ( ) from the top right corner of the screen and selecting *View Source*. This button is available on the Edit Web Content screen, after you've created your web content article.

An XML version of an article is essential when creating content for themes using the Resources Importer. If you'd like to learn more about importing web content with a theme, visit its dedicated tutorial.

The bottom menu of the New Web Content form provides options for customizing your web content.

Structure and Template: lets you customize the web content article's structure and template. To learn more about web content structures and templates, visit the Designing Uniform Content section.

Small Image: sets the image that is used for the web content article's previews. For example, when viewing an article in the Web Content library, the small image is displayed as the article's icon.

Metadata: let you set the organizational hierarchy of the web content article by selecting tags, categories, and priority. To learn more about tags and categories, visit the Organizing Content with Tags and Categories section.

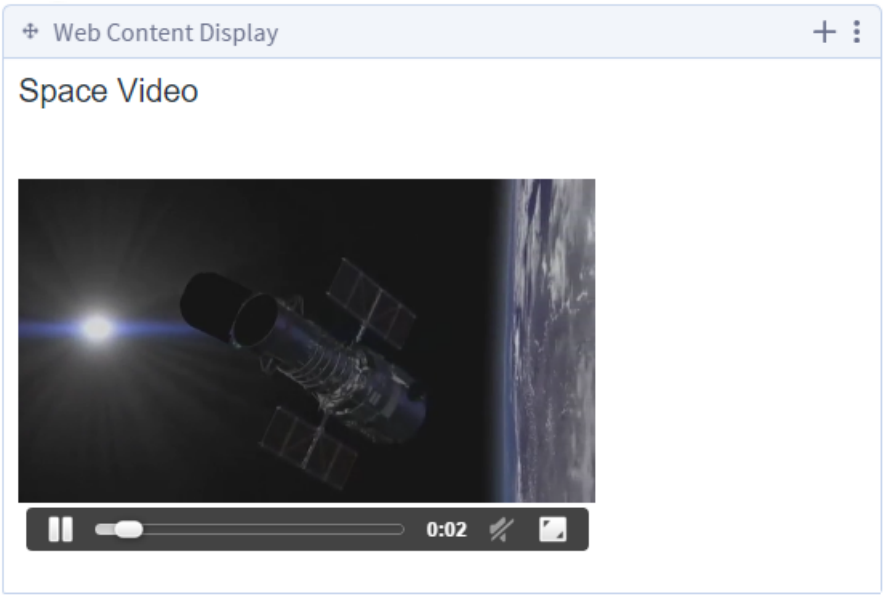


Figure 9.5: If you've installed and enabled Xuggler from the *Server Administration* → *External Tools* section of the Control Panel, you can add audio and video to your web content!

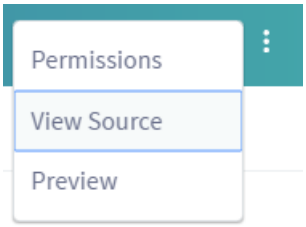


Figure 9.6: The *View Source* button is available from the *Options* button.

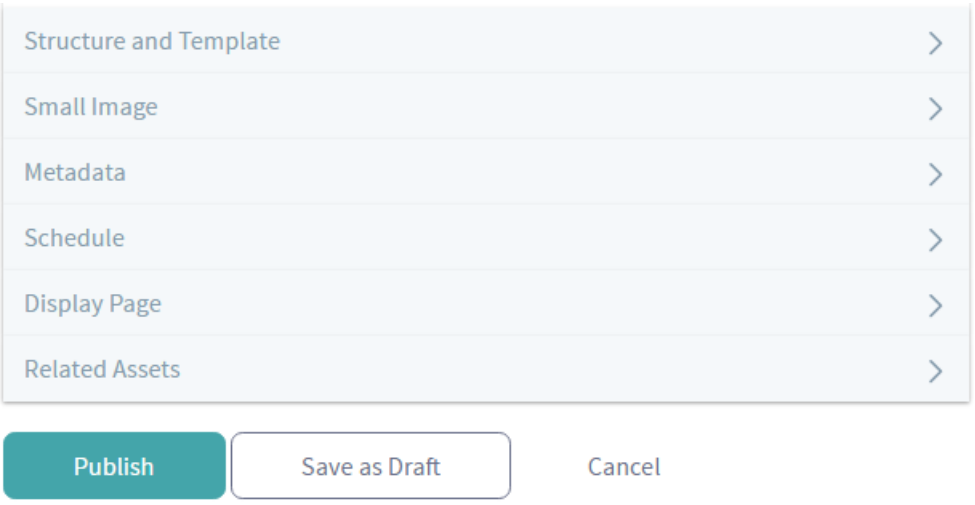


Figure 9.7: New web content can be customized in various ways using the menu located below the WYSIWYG editor.

Schedule: customizes the date and time your content publishes and/or expires. To learn more about scheduling content, visit the Scheduling Web Content Publication section.

Display Page: lets you determine where the web contents are displayed when linked from other pages. The Canonical URL can be used here. The Canonical URL is unique for articles that redirect the visitor to the article's default display page.

Imagine you have a newspaper with a sports section and a technology section. You add a Sports page and a Tech page to your site, each one with a specific banner and look and feel. You want the articles to appear in the appropriate pages, but you know in Liferay DXP, articles are not related to pages. You can add an article as often as you like in different web content display apps or in configured Asset Publishers. But if you have a *View in context* link, where will you show your article? This is where you'd use a default display page. Articles that have a default display page defined are shown with other related articles in the same display page.

Imagine you have 100 sports articles and 100 tech articles. Instead of needing to create a page for each article to show it, you can have only one sports page and one tech page, and can show all articles in one place in a consistent fashion. You'll work through an example of creating a display page in the Creating a Display Page sub-section.

Related Assets: lets you determine content relationships between the web content article and other assets in your Liferay instance, even if they don't share any tags and aren't in the same category. You can connect your content to a Blogs Entry, Message Boards Message, Web Content, Calendar Event, Bookmarks Entry, Documents and Media Document, Wiki Page, etc. To learn more about defining content relationships and publishing links to those related assets, visit the Defining Content Relationships section.

Permissions: customizes who has access to the content. By default, content is viewable by Anyone (Guest Role). You can limit viewable permissions by selecting any Role from the drop-down or in the list. Additionally, Liferay DXP provides the ability to customize permissions in more detail. Select the *More Options* link below the drop down button and you'll find the different activities you can grant or deny to your web content article.

While you can set permissions here, they are ignored unless you activate Web Content Article permissions in your System Configuration:

1. Go to the *Control Panel → Configuration → System Settings*.
2. Search or browse for *Web Content (Default Settings for All Instances)*.
3. Check the box labeled *Article view permissions check enabled*.
4. Click *Save*.

Once it is activated, any permissions you set in the article's configuration are checked before displaying the article.

Version Note: This property is only available with Fix Pack de-13 installed. If you do not have the latest fix packs installed, set the `journal.article.view.permission.check.enabled=` to `true` in your `portal-ext.properties` file and restart Liferay to activate the permission check.

Before you display your web content, you'll learn how to localize it to cater to different language speaking users.

Localizing Web Content When you create a new piece of web content, you have the ability to choose a default language. First, you'll need to change the system configuration to enable the option to change the default language. Go to the *Control Panel → System Settings*. From *System Settings* scroll through the available

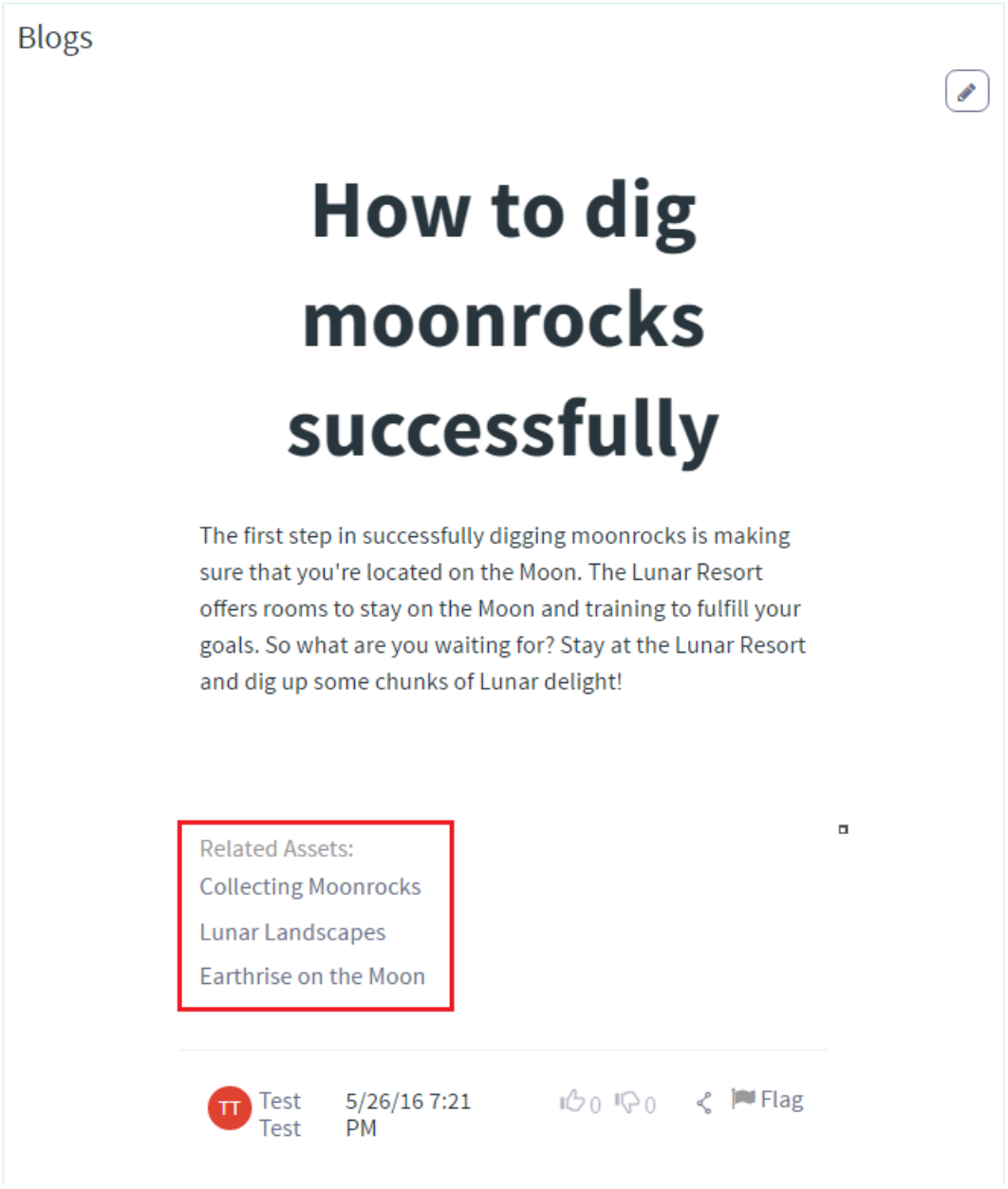


Figure 9.8: This blog entry has links to three Related Assets: one web content and two message board entries.

configurations or use the search bar to find *Web Content Administration*. From there you can check the box to enable *Changeable Default Language* and save your configuration.

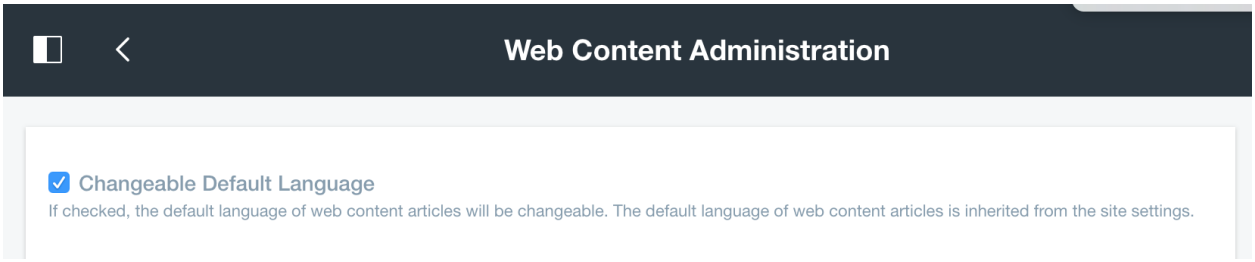


Figure 9.9: This blog entry has links to three Related Assets: one web content and two message board entries.

After you enable changes to the default language, you'll see options at the top of the New Web Content Screen to change the default language and add a translation. If you click *Change*, you can select your default language from a large number of languages Liferay DXP supports.

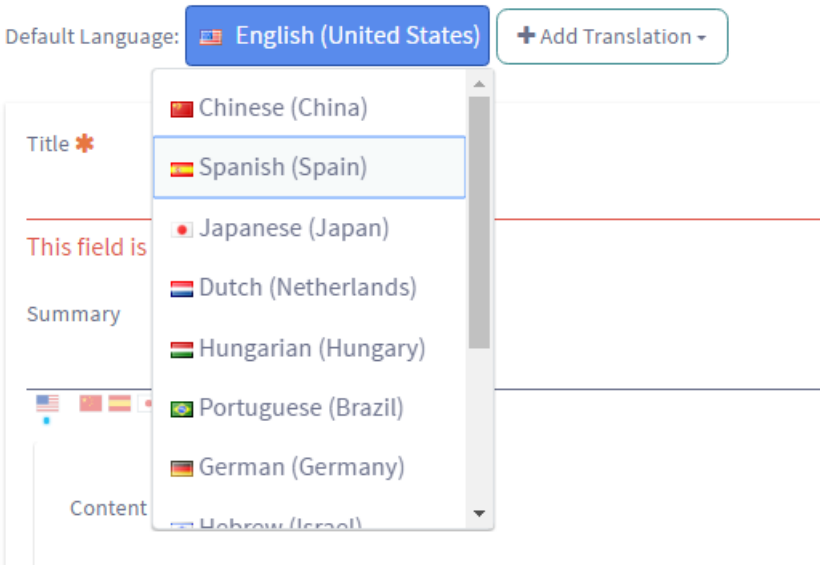


Figure 9.10: : You have many translation languages to choose from for your web content.

After you click *Add Translation*, you can select a language by scrolling through the list. When you select a language, an *Available Translations* list is rendered and the language you selected is highlighted. The new web content form enables you to translate the original web content into the selected language. Once you are done with the translation, click *Publish* and the translation is added to the list of *Available Translations*.

Note: To view localizable fields in a given language, you must have your Portal set to that language. This includes friendly URLs for the web content as well. When you navigate to the localized friendly URL (e.g. `http://localhost:8080/web/guest/-/espanol`), the web content is always displayed in the current language. You can change the language with the Language Selector app.

You can modify the language translation list by inserting `locales.enabled=` followed by your preferred languages in your `portal-ext.properties` file. For example, `locales.enabled=ar_SA,nl_NL,hi_IN` offers *Arabic (Saudi Arabia)*, *Dutch (Netherlands)*, and *Hindi (India)*.

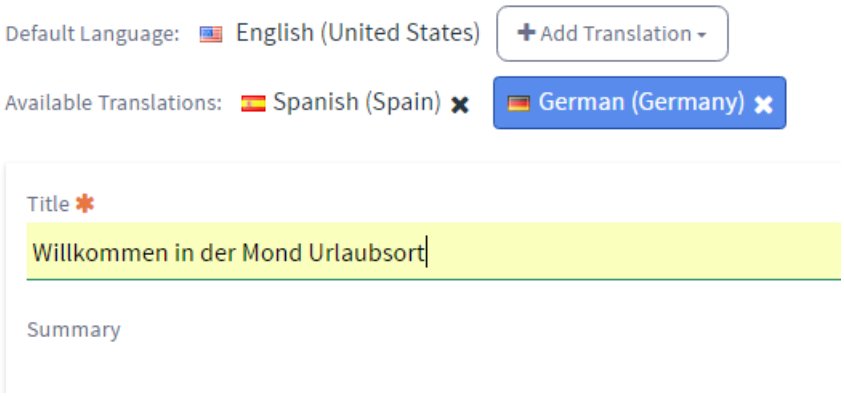


Figure 9.11: : The Available Translations list lets you easily survey the current translations for the article.

Warning: If you switch your site's default language (e.g., via friendly URL), but do not have the necessary translations for localizable fields, your site's language values will be used from the old default language. Therefore, you should change the default language of your site *only* when you have translated values for all localizable entities. Otherwise, you may not be in control of what language is displayed in your Liferay instance.

The ability to completely delete a translation in one step is also available. Instead of disabling a translation or having to go through a multistep process to remove it, you can select the Delete button (X) next to the translation to delete it.

When you create a new web content structure, each field you create has a *Localizable* checkbox displayed next to it. This enables you to control what can and can't be changed in the translation process. For example, if you don't want images or content titles to be changed when the content is translated, you can make sure those fields aren't listed as localizable. When you follow the steps above to localize content, only fields within the structure that had the *Localizable* box checked appear within the translation window.

Next, you'll begin creating a display page to show your web content.

Creating a Display Page There are two ways of creating a display page. You can use a *Content Display Page* template, which automatically creates everything you need, or you can create one manually. The Content Display Page template is found under *Page Templates* in the Sites section of the Control Panel.

To create a display page manually, add an Asset Publisher to a page. Then make it the Default Asset Publisher for the page. This defines this Asset Publisher as the one that displays the content if several Asset Publishers are on the same page. Set this up by clicking *Configuration* on your Asset Publisher. Under the *Setup* tab, navigate to *Display Settings* and check the checkbox labeled *Set as the Default Asset Publisher for This Page*.

Once you've given an article its default display page, links to the article redirect the user to its default display page. To see how this works, add an Asset Publisher to another page, like the Home page of the newspaper, and configure it to *View in Context*. This setting is found in the *Asset Link Behavior* menu under Display Settings. If you click on the link, you'll be redirected to the Default Display Page of the article.

You now see that the link looks something like this:

```
www.lunar-resort.com/lunar-article
```

This is an example of a canonical URL, and it's a nice enhancement for Search Engine Optimization (SEO) because the article's URL becomes the page URL. To a search engine that's crawling your site, this means

that the location of your article never changes. Also, if you decide to use the content on another page in the future, the article is still available at this URL. This feature is used in search results, in related assets, and in Asset Publishers. For more information on Liferay's Display Pages, see the Content Display Pages article.

For this piece of web content, you don't need to change anything. After you're finished with permissions, click *Save as Draft*. This saves the content in draft form. Once you're satisfied with your changes, select *Publish*. This makes the content available for display, but you still have some work to do to enable users to see it. In Liferay WCM, all content resides in a container, which is the Web Content Display app. You'll look at how it works next.

Displaying Web Content

Now that you've created and published your first piece of web content for the Lunar Resort, it's time to display it. First, add the *Web Content Display* application to your Welcome page by selecting the *Add* button ( ) from the top Control Menu and selecting the *Applications* tab. In the search field, type *Web Content Display*.

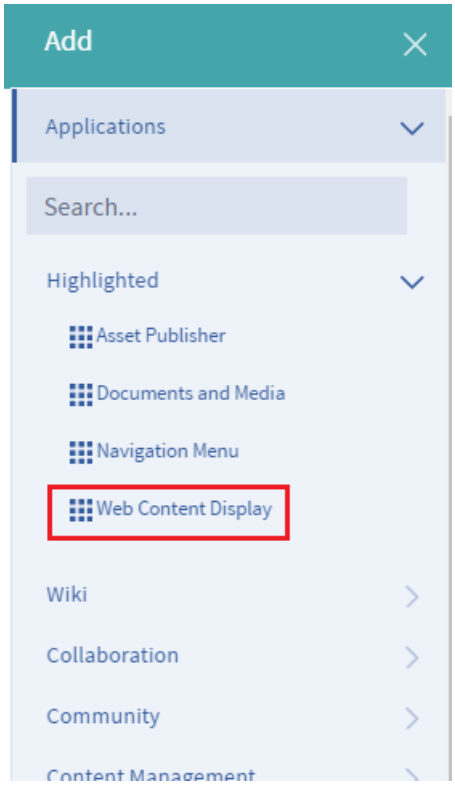


Figure 9.12: : Add the Web Content Display app to a page to begin displaying your new web content article.

Once the application appears, drag it to the position on the page where you want your content to appear. You can have as many Web Content Display apps on a page as you need, which gives you the power to lay out your content exactly the way you want it.

To add existing web content, click the *Select Web Content* button on the lower left of the app. Click the *Select* button from the menu to choose the article you'd like to display. You have several options here.

Naturally, if your content appears in the list, you can simply select it. If there is lots of published content available, you could search for the content by title, description, user name, or site (click the dropdown arrow to see all the options).

Once you've selected the web content article, you're able to choose the User Tools and Content Metadata to be published in the Web Content Display app. These two entities have the following options to choose from, by default:

- **User Tools**
  - *Translations*
  - *Print*
- **Content Metadata**
  - *Related Assets*
  - *Ratings*
  - *Comments*
  - *Comment Ratings*

One of the many options is *Translations*, which shows the available locales for your content. If you're working on the page for a particular language, you can select the translation of your content that goes with your locale. To learn more about translating your content, visit the Localizing Web Content sub-section.

If you have enabled OpenOffice.org integration with your Liferay instance, you can also enable document conversion for your content. This gives your users the ability to download your content in their format of choice. This is especially handy if you are running a research or academically oriented site; users can very quickly download PDFs of your content for their research projects. These conversion options will be available under the *User Tools* list.

Note: To enable OpenOffice integration in your Liferay instance, navigate to the Control Panel → *Configuration* → *Server Administration* → *External Services* and select the *Enabled* checkbox for enabling OpenOffice integration.

Note that you also have other options, such as enabling a Print button, enabling ratings so users can rate the content, enabling comments, and enabling ratings on comments.

The Print button pops the content up in a separate browser window that contains just the content, without any of the web site navigation. This is handy for printing the content. Enabling ratings shows one of two ratings interfaces Liferay DXP has: five stars or thumbs up and thumbs down. This can be set globally in the `portal-ext.properties` file. See the Properties Document for more details about this.

Enabling comments creates a discussion forum attached to your content which users can use to discuss your content. Enabling ratings on comments gives your users the ability to rate the comments. By default, guests are not allowed to leave comments on web content. If you'd like to allow guests to comment on your web content article, navigate to the Control Panel → *Users* → *Roles* and select *Guest* → *Define Permissions*. From the left menu, select *Site Administration* → *Content* → *Web Content*. The navigate down to the Web Content Article heading and select the *Add Discussion* checkbox. Guests are now able to post comments on your web content article!

You may decide you want one, some, or none of these features, which is why they're all implemented as simple selector buttons to be enabled or disabled at need. Once you've selected the features you want to include in your Web Content Display spp, click the *Save* button. You can now close the configuration window.

To publish new content, select the *Add* icon () from the app's top panel and select the type of article you'd like to add (e.g., Basic Web Content). This launches the same full-featured editor you've already seen in the Menu, which lets you add and edit content in place as you are working on your page.

This is another example of the flexibility that Liferay DXP offers. At times, you may want to add content directly into the Web Content Display app of the page you're managing, especially if you are in the process

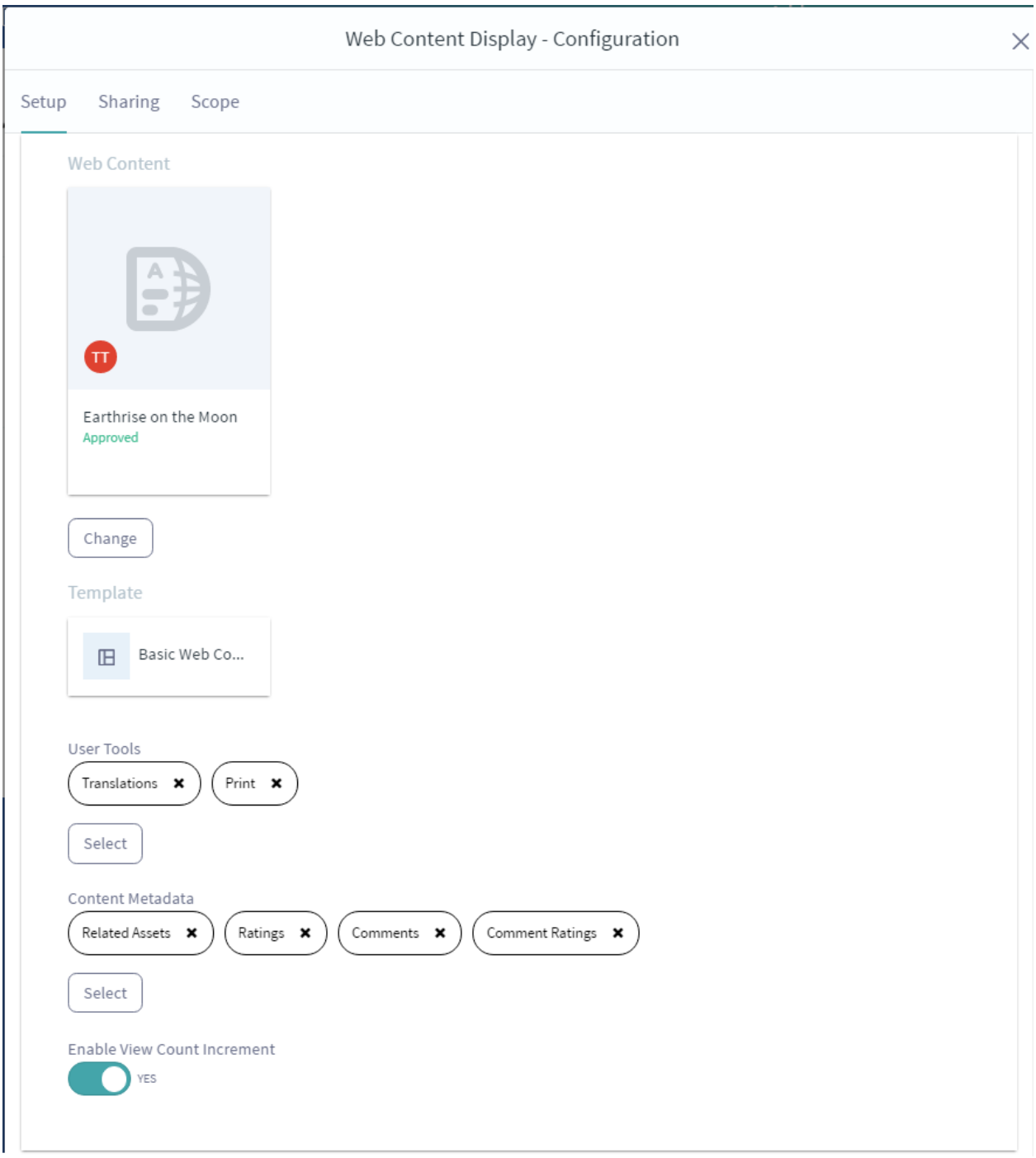


Figure 9.13: : Publishing web content is a snap. At a minimum, you only have to select the content you wish to publish. You can also enable lots of optional features to let your users interact with your content.

of building the page. At other times, you may want to navigate to Site Administration to create content, because at that moment you're more concerned with the creation of the content and not where the content will later be displayed. Liferay WCM supports both processes.

Editing content that's already been published is just as easy as creating new content is. Whether content has been displayed or not you can edit it from Site Administration. To edit content,

1. Go to *Site Administration* → *Content* → *Web Content*.
2. Click the *Options* button ( ) next to the article that you want to edit and select *Edit*.

   This launches the WYSIWYG editor and from there you can make any necessary changes.

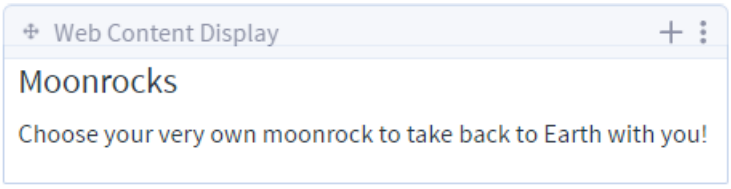


Figure 9.14: : You can select and edit an article, or edit its template directly from the Web Content Display app.

There are instances where you've edited your web content article many times, and you'd be interested in viewing the article's evolution. To view an article's history, navigate to *Web Content* from the Menu. Then select the article's *Actions* icon ( ) and select *View History*. From this menu, you can view all the article's versions and modified/display dates. Another cool feature is the web content Diff tool, which lets you compare versions of the article and highlight the differences between the two. Of course, you must have more than two versions of the article for this feature to be available. Click the *Actions* icon again next to a version of the article you'd like to compare and select *Compare to...*. Then select the other article you want to compare. The tool provides color coded highlighting to emphasize additions and deletions between the two articles.

When you publish updates to a web content article that's already being displayed somewhere in your Liferay instance (e.g., in a Web Content Display app or an Asset Publisher app), the content is immediately updated (unless, of course, you have a workflow enabled, which is discussed in greater detail in the Using Workflow section). It makes no difference whether you edit it from a Web Content Display app, from the Asset Publisher, or from the Site Administration interface.

Note: If you're using a mobile device or tablet and you'd like to view your page the way your users will see it (i.e., without all the app controls and icons), go up to the top Control Menu and select the *Toggle Controls* icon ( ). This makes all those extra controls you see as a Liferay administrator disappear. You'll also notice the icon is crossed out when the Toggle Controls are disabled. If you need to use the controls again, just select the icon again to return to the original format.

This button is not displayed when viewing the page from a desktop computer. This is because apps displayed from a desktop computer hide their controls by default, and can be rendered by hovering over the app. To test out your page from different devices, select the *Simulation* button ( ) from the right corner of the top Control Menu.

As an administrator, you may want to monitor what changes are being made to your site's web content without implementing a workflow process. To keep tabs on what's going on with your site's web content,

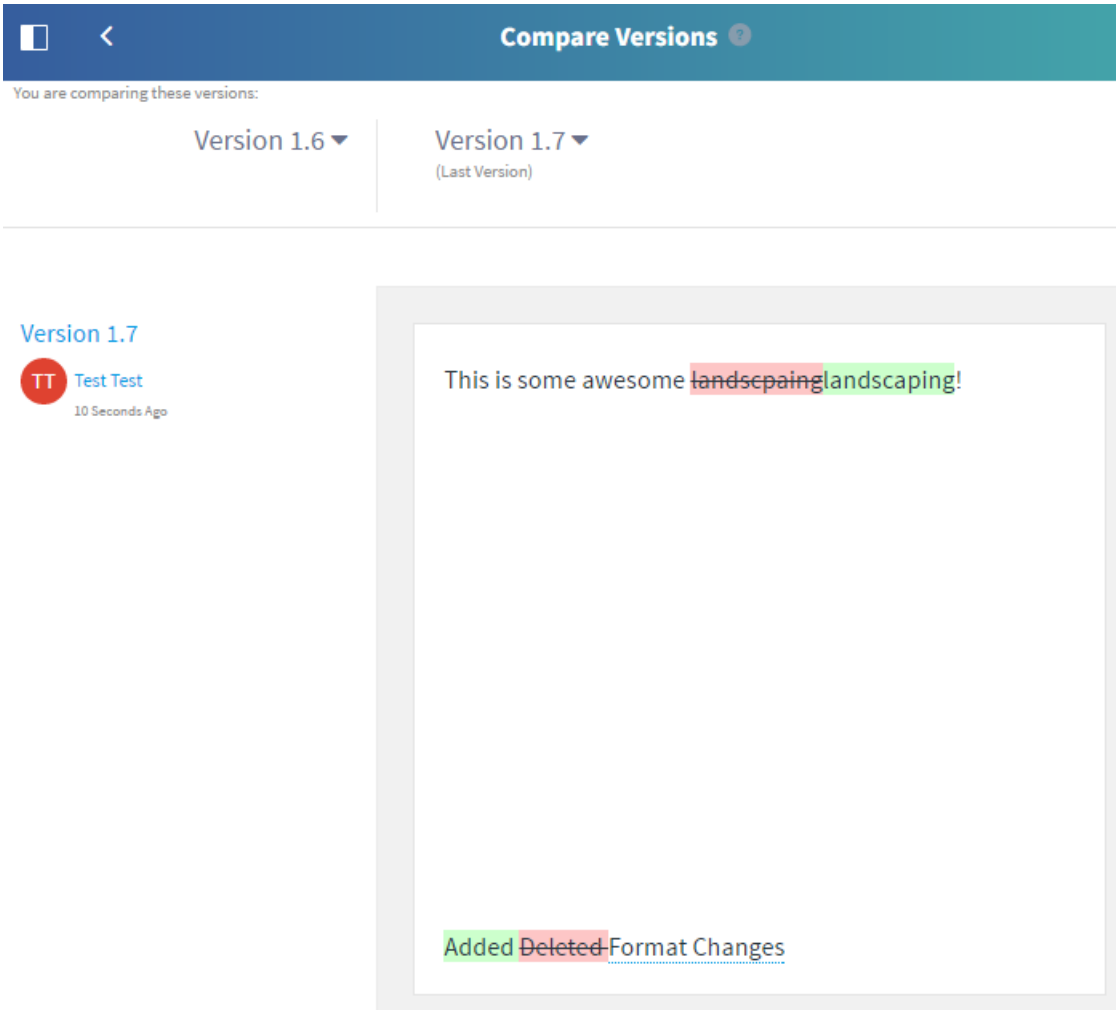


Figure 9.15: : Comparing web content articles is a great feature to use during the Workflow process.

you can subscribe to articles and folders. To do this, select the checkbox next to the web content entities you'd like to monitor. Then click the *Information* icon ( ) and select the *Subscribe* icon ( ). Now whenever a web content article or folder is modified, you'll receive an email to your account's configured email address notifiying you of a change. To learn more about configuring your email in Liferay DXP, visit the Instance Settings section. You can navigate to your Web Content menu's *Options* icon ( ) and select *Configuration* to modify your Web Content email notification settings.

That's pretty much all there is to simple content creation. Whole sites have been created this way. But if you want to take advantage of the full power of Liferay DXP's WCM, you'll want to use structures and templates. You'll cover these topics next.

9.2 Designing Uniform Content

If you've ever launched a web site, you know that as it grows, you can experience growing pains. This is the case especially if you've given lots of people access to the site to make whatever changes they need to make. Without preset limitations, users can display content in any order and in any manner they desire (think huge, flashing letters in a font nobody can read). Content can get stale, especially if those responsible for it don't

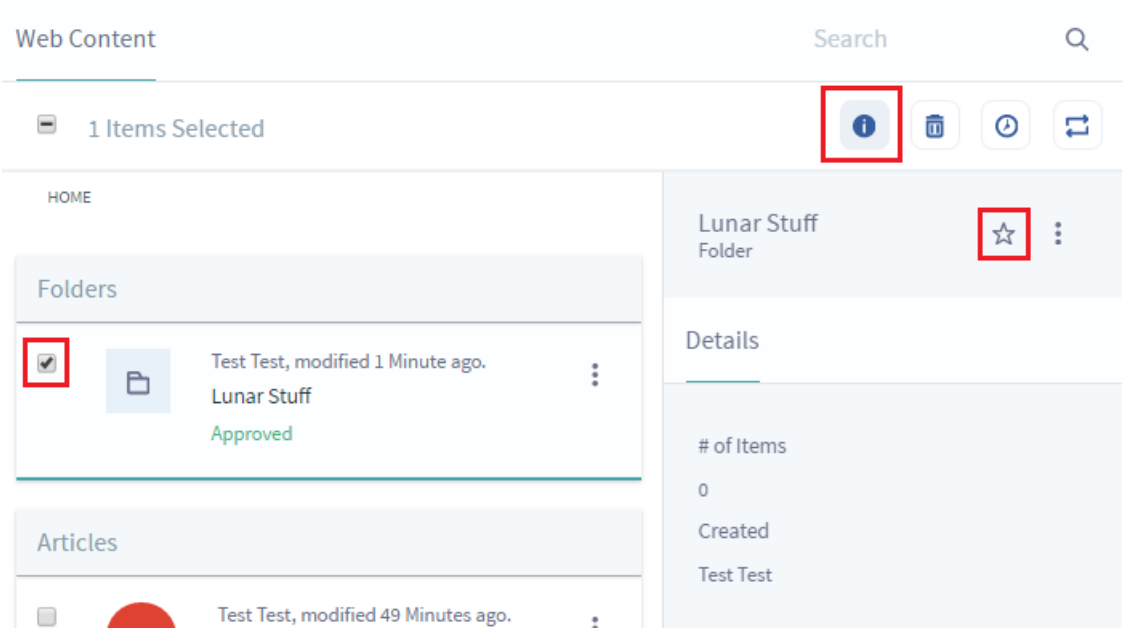


Figure 9.16: : Click the Subscribe icon in the web content entity's *Information* menu to begin receiving web content notifications.

maintain it like they should. And sometimes, content is published that should never have seen the light of day.

!PVideo Thumbnail

Thankfully, Liferay WCM helps you handle all of those situations. You can use *Structures* to define which fields are available to users when they create content. These can be coupled with *Templates* that define how to display that content. Content won't get stale, because you can take advantage of the Scheduling feature to determine when content is displayed and when it's removed. Additionally, you can configure Liferay DXP's built-in Workflow system to set up a review and publishing process so only what you want winds up on the live site. Liferay DXP gives you the management tools you need to run everything from a simple, one-page web site to an enormous, content-rich site.

All of this starts with structures.

Creating Structured Web Content

Structures are the foundation for web content. They determine which fields are available to users as they create new items for display. Structures not only improve manageability for the administrator, they also make it much easier for users to quickly add content.

For example, say you're managing an online news magazine. All your articles need to contain the same types of information: a title, a subtitle, an author and one or more pages of text and images that comprise the body of the article. If Liferay DXP only supported simple content as has been described above, you'd have no way to make sure your users entered a title, subtitle, and author. You might also get articles that don't match the look and feel of your site. If titles are supposed to be navy blue but they come in from your writers manually set to light blue, you need to spend time reformatting them before they are published.

Structures give you the ability to provide a format for your content so your users know what needs to be entered to have a complete article. Using structures, you can provide a form for your users which spells out exactly what is required and can be formatted automatically using a template.

You create a structure by adding form controls such as text fields, text boxes, text areas (HTML), check boxes, select boxes and multi-selection lists. Also, you can add specialized, Liferay-specific application fields such as an Image Uploader and Documents and Media right onto the structure. Furthermore, you can move the elements around by dragging them where you want them. This makes it easy for you to prototype different orders for your input fields. Additionally, elements can be grouped together into blocks which can then be repeatable. Template writers can then write a template which loops through these blocks and presents your content in innovative ways, such as in sliding navigation bars, content which scrolls with the user, and more.

Next you'll take a look at how you can create and edit structures through the Manage Structures interface.

Editing Structures Go to your site's Site Administration menu and select *Web Content* from the Content section. The first way to access the Manage Structures interface is by navigating to the *Options* icon ( ) in the top right of the page and selecting *Structures*. This opens a popup showing all the web content structures that exist in your currently selected scope. Here, you can add new web content structures, edit existing ones, manage the templates associated with a structure, edit the permissions of a structure, and copy or delete structures.

Copying web content structures can be useful if you'd like to create a new web content structure that's similar to an existing one, but you don't want to start from scratch. Liferay DXP generates a unique portal ID for the copied structure, but every other attribute of the copied structure, including the name, is the same as that of the original. Once you've copied a web content structure, you should enter a new name for it to avoid confusing it with the original. When you copy a web content structure, you'll be prompted to choose whether to copy any detail templates or list templates associated with the structure. For information on detail templates and list templates, please refer to the Using Web Forms and Dynamic Data Lists section.

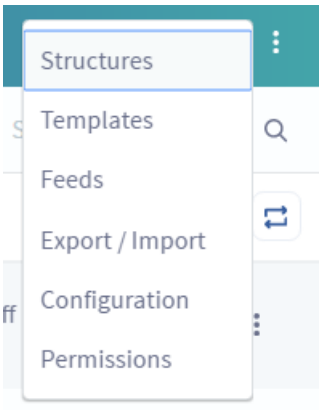


Figure 9.17: You can access the Manage Structures interface by clicking the Options icon → *Structures* from the Web Content page.

The second way to access the Manage Structures interface is directly from the web content article menu. Click *Add* → *Basic Web Content* from the Web Content page to add another piece of content to your Liferay instance. Instead of going right for the content, this time you'll first create a structure. To access the Manage Structures interface, simply click on *Structure and Template* in the bottom dropdown menu and click *Select* for the *Structure* heading. You'll notice there is a *Basic Web Content* structure and template available by default. This structure and template are used automatically if a custom structure and template are not added. You have the option of editing the default structure and template, if desired. To create a new structure in your chosen scope, simply click on the *Add* button ( ) in the Manage Structures popup.

It's very easy to create and edit structures: all you have to do is drag elements into the structure and then give them names. For instance, select the *Text* element and drag it onto the structure. You can do the same with any of the elements. To remove it from the structure, simply select the *Delete* icon ( ) in the upper right corner of the element. You also have the ability to duplicate the element, which can be done by selecting the *Duplicate* button ( ). You'll learn about the *Configuration* ( ) button later.

Web content structures also have the capability of inheriting characteristics from other structures. When a parent structure is configured, the child structure inherits the parent's fields and settings. Using this feature is helpful when you want to make a similar structure to one that already exists. For example, if you'd like to create an in-depth Lunar Resort sports article in addition to a regular Lunar Resort sports article, you can simply inherit the characteristics of the regular article and only add additional fields to the more in-depth article. When the in-depth article is configured, it will display its parent's fields in addition to its own fields.

Note: In some instances, there can be more than one structure with the same `structureKey`. For example, this can happen when exporting a global structure and then importing it back into a site. This scenario would have global and site scoped structures with identical `structureKeys`. If this happens, you can no longer use the global structure. This is because Liferay DXP is configured to follow a specific hierarchy when choosing structures with the same `structureKey`: *current site > parent site > global scope*.

The WebDAV URL feature is available for web content structures and templates so users could upload and organize resources from both a web interface and the file explorer of their desktop operating system. With the WebDAV URL, site administrators are capable of adding, browsing, editing, and deleting structures and templates on a remote server. After you complete your structure, you can access the WebDAV URL by re-opening the structure or template and clicking the *Details* section. If you'd like the see WebDAV in action, visit the WebDAV Access section.

Note: Some operating systems require a WebDAV server to be class level 2 (i.e., to support file locking) before allowing files to be read or written. The Documents and Media library uses a class level 2 WebDAV server but Web Content structures and templates do not. This means that Liferay DXP's Document and Media library supports WebDAV file locking but Web Content structures and templates do not. However, on operating systems which require WebDAV servers to be class level 2, it's possible to avoid the restriction by using third-party WebDAV clients (e.g., Cyberduck).

Another method to edit your structure is switching to *Source* mode and manually customizing your structure by editing its XML file. You'll notice by default the *View* mode is selected. Click the *Source* tab to switch to Source mode. This method is for the more experienced developers.

Take a moment to add, delete, and rearrange different elements.

Liferay DXP supports the following fields in structures:

Boolean: Adds a checkbox onto your structure, which stores either `true` (checked) or `false` (unchecked). Template developers can use this as a display rule.

Date: Adds a preformatted text field that displays a convenient date picker to assist in selecting the desired data. The format for the date is governed by the current locale.

Decimal: Similar to *Number*, except that it required a decimal point (.) be present.

Documents and Media: Adds an existing uploaded document to attach to the structure. Also has the ability to upload documents into the Document Library.

Geolocation: Adds a map that displays a configured location. The geolocation system can work in two ways: letting the system know your current location (especially useful on mobile devices) and giving the user

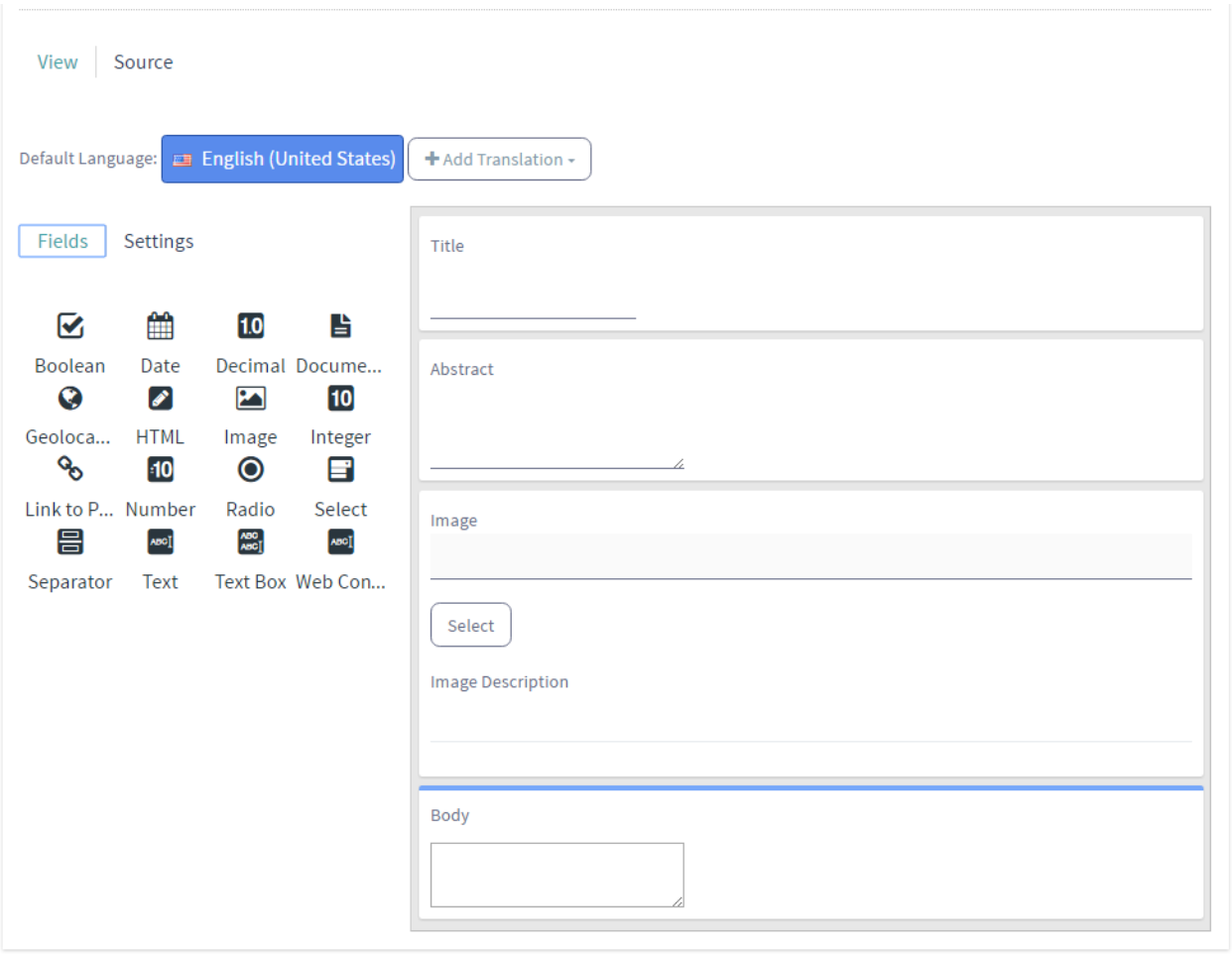


Figure 9.18: The structure editor gives you many options to customize your Web Content.

directions to a concrete place.

HTML: An area that uses a WYSIWYG editor to enhance the content.

Image: Adds the browse image application into your structure. You have the option of selecting an image from the Documents and Media library or to upload an image from your computer's storage. If uploading an image from your personal computer to the web content article, it is only available for that article.

Integer: Similar to *Number*, except that it constrains user input to non-fractional numbers.

Link to Page: Inserts a link to another page in the same site.

Number: Presents a text box that only accepts numbers as inputs, but puts no constraints on the kind of number entered.

Radio: Presents the user with a list of options to choose from using radio button inputs.

Select: Presents a selection of options for the user to choose from using a combo box. Can be configured to allow multiple selections, unlike *Radio*.

Separator: Adds a line separator between fields, useful for organization purposes.

Text: Used for items such as titles and headings.

Text Box: Used for the body of your content or long descriptions.

These fields provide all you need to model any information type you would want to use as web content. Liferay customers have used structures to model everything from articles, to video metadata, to databases

of wildlife. You're limited only by your imagination. To fuel that imagination, you'll look more closely at field settings.

Editing Field Settings When creating a new structure, it is essential that you set variable names. Template writers can use these variables to refer to elements on your form. If you don't set variable names, Liferay DXP generates random variable names and these can be difficult for a template writer to follow. For example, consider a field called *Author*. You might create this field in your form but the underlying variable name in the structure might look something like `TextField4882`. The template writer needs to create markup for your structure and place the Author field in a certain spot in the markup. How will he or she know which field is Author when they're all named randomly?

To solve this problem, all you need to do is set a variable name for each field as you add it to your structure. Go ahead and do this now. In your structure, add an element *HTML*. To change its field label and variable name, you'll need to access the field's settings. Hover over the field and select the *Configuration* icon ( ) that appears in the upper right corner. Change the *Field Label* value to *Instructions* and the *Name* value (variable name) to `Steps`. Now your template writer has a variable by which he or she can refer to this field.

Here's a list of all the configurable settings available for a structure's fields:

Type: Lists the type of field placed in the definition. This is not editable but is available to reference from a template.

Field Label: Sets the text that can be displayed with the field. This is the human-readable text that the user sees.

Show Label: Select *Yes* to display the Field Label.

Required: Select *Yes* to mark the field required. If a field is required, users must enter a value for it in order to submit content using this structure.

Name: The name of the field internally, automatically generated. Since this is the variable name that you can read the data from in a template, you should give a more memorable name here.

Predefined Value: Specifying predefined values for structure forms is a way to specify defaults. When a user creates a new web content article based on a structure that has predefined values for various fields, the predefined values appear in the form as defaults for those fields.

Tip: Each field can have a small help icon, with a tooltip attached that displays helpful information. If you would like to provide text for the tooltip you may enter it here.

Indexable: Select *Yes* to enable Liferay DXP to index your field for search.

Localizable: Select *Yes* to enable Liferay DXP to localize your field.

Repeatable: Select *Yes* to make your field repeatable. Your users can then add as many copies of this field as they like. For example, if you're creating a structure for articles, you might want a repeatable Author field in case you have multiple authors for a particular article.

Multiple: Select *Yes* to enable a multi-selection list (only available for Select).

Options: Changes the options available for selection. You're able to add and remove options as well as edit each individual option's display name and value (only available for Radio and Select).

Style: Changes the line separtor's style (only available for Separator).

For the Lunar Resort structure, type something in the *Tip* field that helps users know what to put into the Body element (example: *This is an HTML text area for the body of your content*). Now, when users hover over the Help icon near your title, your tip is displayed.

Structure Default Values Structure Default Values let you easily fill in values which will be repeated for content created from that structure. You can use Structure Default Values to set defaults for Liferay's standard asset fields (like tags, categories, and related assets) and the content of the structure fields, as well as setting a default template for displaying the structure data.

Returning to the newspaper scenario again, assume you want all sports articles to have the same display page (sports page), the same categories, or the same set of tags. Instead of adding them for each article or wondering if your users are adding them to every web content article, you can add these characteristics once for every sports article by creating default values for the structure. Creating default values is not part of creating a new structure, so make sure you have an existing structure.

To edit a structure's default values, go to *Web Content* in the Content section of Site Administration and click the *Options* icon ( ) → *Structures* to see the structures list. Find the *Actions* button ( ) for the desired structure and select *Edit Default Values* from the menu to view a window like the one below. This form allows you to manage the structure settings.

Every new web content you create with this structure is preloaded with the data you inserted. Next, you'll learn about assigning permissions.

Assigning Permissions Setting permissions on structures is done using the same procedure as permissions everywhere else in Liferay. Most users should not have the ability to edit structures. Structures are coupled with templates, which require some web development knowledge to create. This is why only trusted developers should be able to create structures and templates. Users, of course, should be able to view structures. The *View* permission enables them to make use of the structures to create content.

You can grant or deny permissions based on Roles and this is the recommended way to handle permissions for structures.

Now that you understand what structures are used for, you need to understand the other half of Liferay DXP's web content management system: templates.

Designing Web Content with Templates

Developers create templates to display the elements of the structure in the markup they want. Content can then be styled properly using CSS, because markup is generated consistently by the template when structured content is displayed. In essence, templates are scripts that tell Liferay DXP how to display content in the structure. Any changes to the structure require corresponding changes to the template, because new or deleted fields produce errors on the page. If users enter content into a structure, it *must* have a matching template. You have options, however, for whether you want your template to be permanently linked to your structure. Generic templates are templates that are not tied to a structure, which allows for reusable code that can be imported into other templates. Without a template, Liferay DXP has no idea how to display content which has been created using a custom structure.

You'll look more closely at the types of templates Liferay DXP supports next.

Template Types (FTL, VM, and XSL) Liferay DXP supports templates written in three different templating languages, to support the skill sets of the largest number of developers. This increases the chances you can jump right in and use whichever one you've already used before. If you haven't yet been exposed to any of them, your best bet is FreeMarker or Velocity, as they are less "chatty" than XSL and extremely simple to understand.

FTL (FreeMarker Template Language): FreeMarker is a templating language which could be considered a successor to Velocity. It has some advantages over Velocity for which it sacrifices some simplicity, yet it is still easy to use. If you haven't used any of the template languages before, FreeMarker is recommended: you'll get up to speed the fastest.

VM (Velocity Macro): Velocity is a scripting language that lets you mix logic with HTML. This is similar to other scripting languages, such as PHP, though Velocity is much simpler.

Note: The Velocity template language is deprecated for Liferay DXP 7.0.

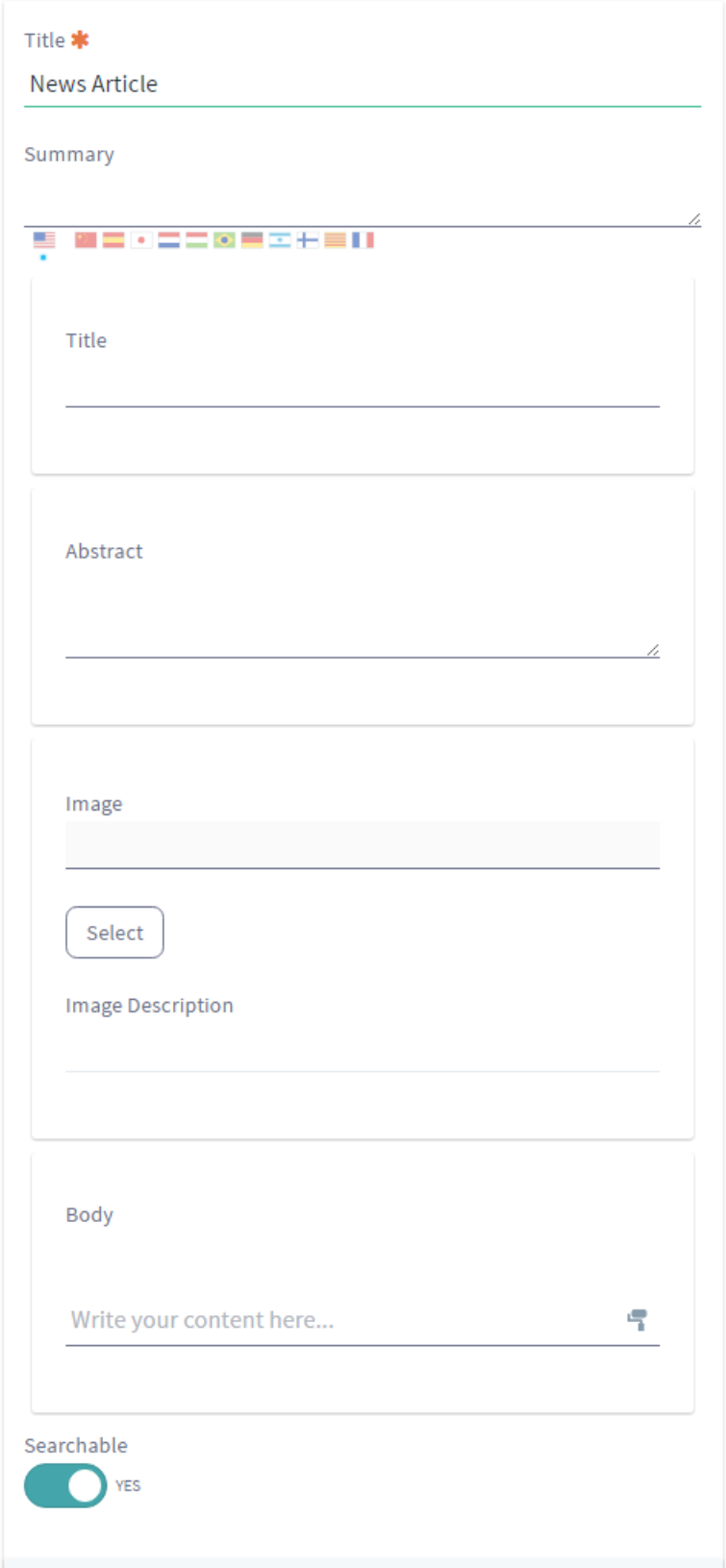


Figure 9.19: You can edit default values via the *Actions* button of the Manage Structures interface.

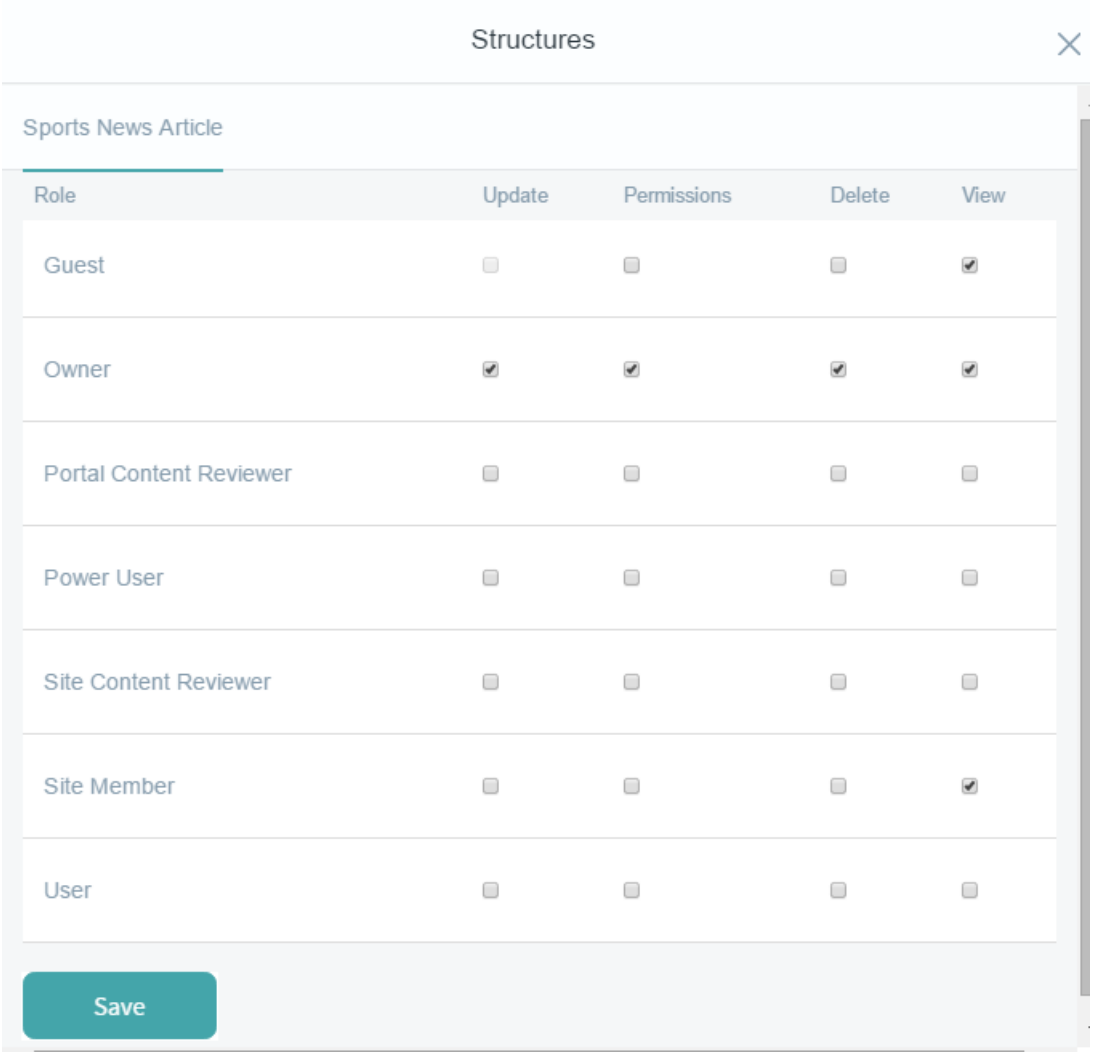


Figure 9.20: You're able to assign structure permissions via the *Actions* button.

XSL (Extensible Style Sheet Language): XSL is used in Liferay templates to transform the underlying XML of a structure into markup suitable for the browser. While it may not be as clean and compact as Velocity or FreeMarker, it's widely used for transforming XML into other formats and it's very likely your developers have already been exposed to it.

Adding Templates Liferay WCM makes it easy to create structures, templates, and content from the same interface. You'll go through the entire flow of how you'd create a structure, link it to a template, and then create content using them both. You'll use FreeMarker for your template and lay out the structure fields systematically to go along with the format you've defined for your content.

1. Go back to the Web Content section of the Site Administration page and click *Add → Basic Web Content*.
2. Select *Structure and Template* from the bottom menu and click *Select* under the Structures heading to access the Manage Structures interface.
3. Click on the *Add* button ( ).

4. Name the structure *News Article* and add the following fields:

Field Type | Field Label | Name |

_——— | ——— | ——— | Text | Title | title | Text Box | Abstract | abstract | Image | Image | image | HTML | Body | body |

5. Click *Save*.
6. In the Manage Structures interface, click *Choose* next to the News Article structure that you created.
7. In the New Web Content form, click *Select* under the Template heading to access the Manage Templates interface.
8. Click *Add*, enter the name *News Article*, and add a description.
9. Make sure FreeMarker is selected as the script language (it's the default).
10. If you've written the script beforehand, you can select *Browse* to upload it from your machine. Otherwise, you can type the script directly into the script editor window.
11. Click *Save*.
12. Click *Choose* next to the News Article template you created.
13. On the New Web Content form, you'll see the Title, Abstract, Image, and Body fields that you defined for the News Article structure. The News Article template should also be selected.
14. Populate the fields and click *Publish* to publish your News Article.

Below is the template script for this structure. It is written in FreeMarker:

```
<#assign renderUrlMax = request["render-url-maximized"]>
<#assign namespace = request["portlet-namespace"]>
<#assign readmore = request.parameters?is_hash && request.parameters.read_more?? && getterUtil.getBoolean(request.parameters.read_more?first, false)>
<h1>${title.getData()}</h1>
<#if readmore>
<p>${abstract.getData()}</p>
<p>${body.getData()}</p>
<#else>
<p>
<img src="${image.getData()}" border="0" align="right">
${abstract.getData()}</p>
<a href="${renderUrlMax}&${namespace}read_more=true">Read More</a>
</#if>
```

This template is small but accomplishes a lot. First, a portlet URL which maximizes the portlet is created. Once this is done, the template gets the namespace of the portlet. This is important to avoid URL collisions with other URLs that might be on the page.

After this, the template attempts to get a request parameter called `read_more`. Whether or not this was successful is the key to the rest of the script:

- If the template got the `read_more` parameter, it displays the abstract and the body below the title (which is always displayed).

- If the template didn't get the `read_more` parameter, it displays the image, the abstract and the link created above, which sets the `read_more` parameter.

Figure 9.21: The initial and expanded views for the Lunar Resort News Article. After clicking *Read More*, you're able to read the full text body.

When this template is rendered, it looks something like this:

Note: During the creation of a web content article, Liferay DXP provides an *Options → Preview* button that gives you the option to preview your article as a final product before publishing. In some instances, the preview does not give an accurate depiction of the web content article. For example, fields provided by the `request` variable are not available because the `request` is not populated until the web content is rendered on a Liferay page. Therefore, the preview of the article would display errors. Use the *Preview* functionality with caution.

Liferay DXP also provides the ability to create generic templates that aren't connected to a specific structure. In previous versions of Liferay, each template had to be associated with a structure. Now, you have options for whether to permanently assign a template to a structure or create a generic template and reuse its code for any structure.

Suppose you have three different Lunar Resort web content articles and structures with similar aesthetics. Instead of creating three different templates from scratch, you can use the same generic template for all three and build off of it. This creates a smarter and more efficient process when creating a multitude of similar web content articles.

You can also embed applications in web content templates. This is a convenient way to ensure that specified apps are always located inside your web content article. Core apps and custom apps, whether instanceable or non-instanceable can be embedded in web content templates. Below are examples of embedding the Currency Converter app in FreeMarker and Velocity:

FreeMarker:

```
<@example_portlet_ext["runtime"] portletName="com_liferay_currency_converter_web_portlet_CurrencyConverterPortlet" />
```

Velocity:

```
$theme.runtime("com_liferay_currency_converter_web_portlet_CurrencyConverterPortlet");
```

Warning: The `theme` variable is no longer injected into the FreeMarker context. For more information about why the theme variable was removed for Liferay DXP 7.0 and suggestions for updating your code, visit the Taglibs Are No Longer Accessible via the theme Variable in FreeMarker breaking change entry.

In addition to embedding applications in templates, you can embed a template within another template. This allows for reusable code, JS library imports, scripts, or macros. The template that you embed should be a generic template with no structure assigned to it. To create a template with no structure, leave the structure field empty when you create the template. To reference a template from within another template, you will need the Template Key.

Below are examples of embedding template in FreeMarker and Velocity:

FreeMarker

```
<#include "${templatesPath}/[template-key]" />
```

Velocity

```
#parse ("$templatesPath/[template-key]")
```

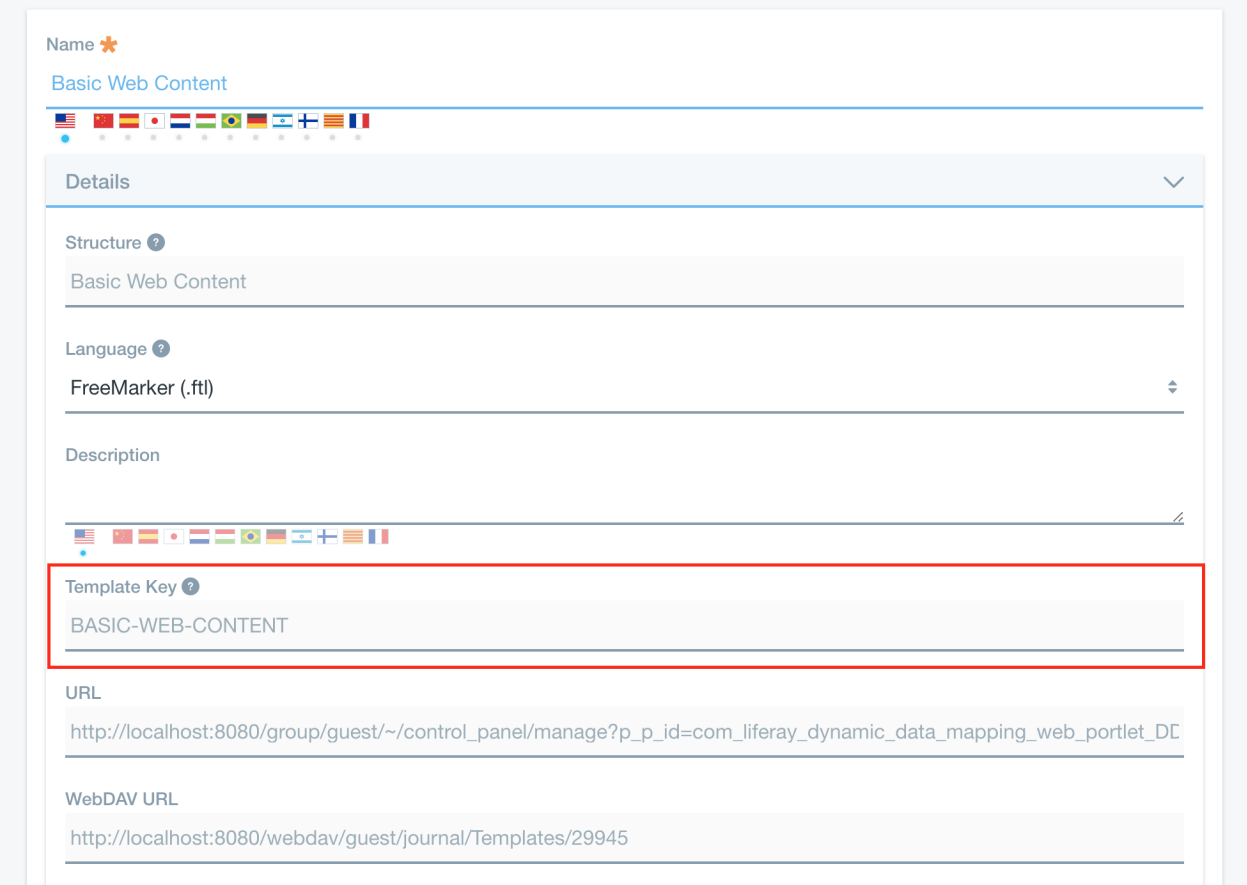


Figure 9.22: You can find the Template Key when view the Edit page for a template..

Liferay's taglibs are also accessible to web content administrators developing in FreeMarker. There is no need to instantiate these taglibs within your FreeMarker template; they're already provided for you automatically. You can access these taglibs by indicating the TLD's file name with underscores. For instance, the above FreeMarker example accessed a tag in the `liferay-portlet-ext.tld` file by specifying `@example_portlet_ext`. This is not available for Velocity users, since Velocity is deprecated for Liferay DXP 7.0.

For cases where you're creating your template within Liferay DXP, you can use the template editor. On the left side of the template editor, you'll notice a palette of common variables used for making web content templates. This is a great reference when creating your template. To place one of the variables onto the template editor, simply position your cursor where you want the variable placed, and click the variable name. If the variable name doesn't give you sufficient information on the variable's functionality, you can hover your pointer over it for a more detailed description.

The interactive template editor is available for the FreeMarker, Velocity, and XSL languages. Depending on which language you select, the variable content changes so you're always adding content in the language you've chosen. Another cool feature for the template editor is the autocomplete feature. It can be invoked by typing *${* which opens a drop-down menu of available variables. By clicking one of the variables, the editor inserts the variable into the template editor.

Note: The `utilLocator`, `objectUtil`, and `staticUtil` variables for FreeMarker and the `utilLocator` variable for Velocity are disabled by default. These variables are vulnerable to remote code execution and privilege

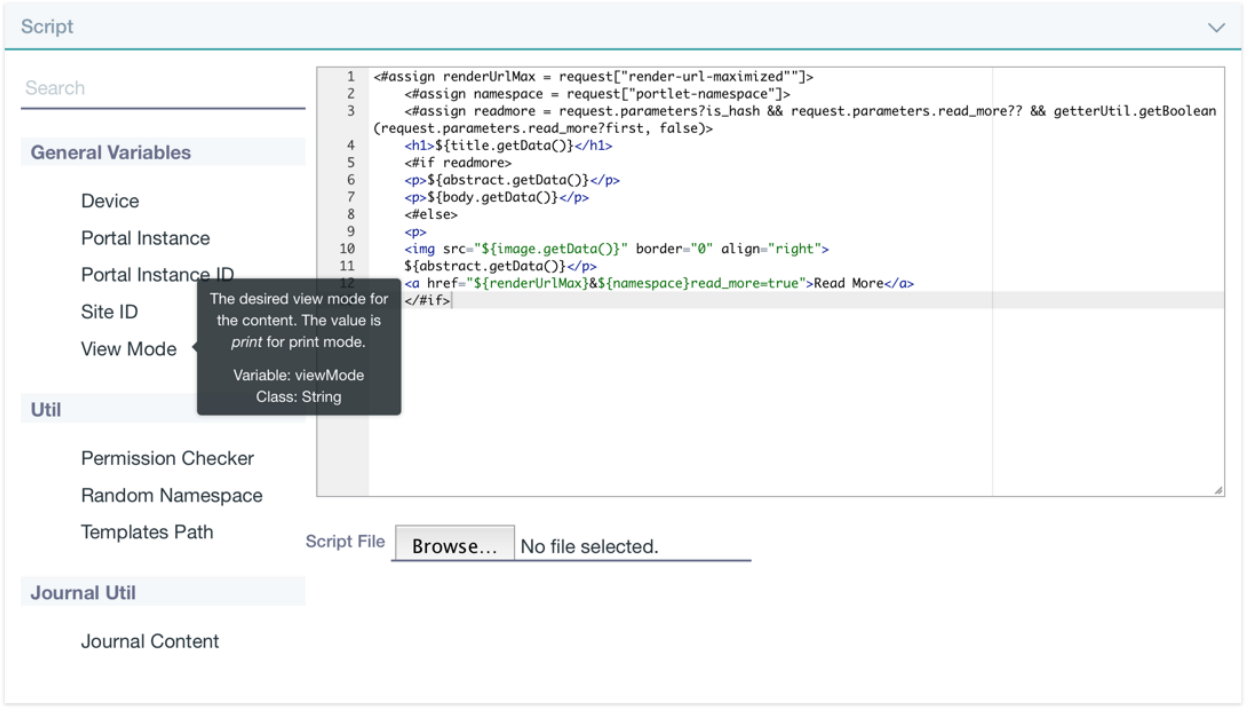


Figure 9.23: You can hover your pointer over a variable for a more detailed description.

escalation, and should be used with caution, if enabled.

After you've saved your template, Liferay DXP provides a WebDAV URL and static URL. These values access the XML source of your structure. You can find these URLs by returning to your template after it's been saved and expanding the *Details* section. For more information on WebDAV and the uses of the WebDAV URL, reference the WebDAV Access section.

Now that you've created a handsome template and know how to use the template editor, it's time to decide who the lucky people are that get to use your new template.

Assigning Template Permissions

Permissions for templates are similar to permissions for structures. As with structures, you only want specific developers editing and creating templates. You may, however, want to make the templates viewable to some content creators who understand the template scripting language but are not directly writing the scripts. You can determine who views and interacts with the template by navigating to the *Options* button ( ) at the top right and selecting *Templates*. Then select the *Action* button ( ) and click *Permissions*.

You can grant or deny permissions based on Roles. For instance, you may create a role with the ability to update the template and create a second role that can both update and delete. Liferay DXP makes it possible to assign permissions based on the roles and responsibilities within your organization.

Whether your site is small and static or large and dynamic, Liferay's WCM enables you to plan and manage it. With tools such as the WYSIWYG editor, structures and templates, you can quickly add and edit content. With the Web Content Display, you can rapidly select and configure what content to display. You'll find that managing your site becomes far easier when using Liferay DXP's Web Content Management system.

!VVideo Tutorial

Chapter 10

Staging Content for Publication

Today's enterprises are generating an enormous amount of content for their users. Liferay DXP provides advanced publishing tools to make the content easily and reliably available for users.

Staging is an important feature of Liferay WCM. The concept of staging is a simple one: you can modify your site behind the scenes and then publish all your updates in one shot. You don't want users seeing your web site change before their eyes as you're modifying it, do you? Liferay DXP's staging environment allows you to make changes to your site in a specialized *staging area*, which is linked to a production environment. Typically the staging site has a more limited audience (e.g., content editors, site administratros, etc.) while the production environment is accessible for most users. When you're finished, you can publish all your site changes at once.

Site administrators can set up their staging environments locally or remotely. With Local live staging, your staging environment and live environment are hosted on the same server, whereas Remote live staging has the staging and live environments on separate servers. You'll learn more about the differences between these two staging environments and how to properly enable them for your Liferay instance.

Liferay DXP also offers the Page Versioning feature. This feature works with both Local Live and Remote Live staging and allows site administrators to create multiple variations of staged pages. This allows several different versions of sites and pages to be developed at the same time. Variations can be created, merged, and published using a Git-like versioning system. In the next section, you'll jump in to see how to enable staging.

Chapter 11

Enabling Staging

Liferay DXP provides site administrators with two different ways to set up staging: Local Live and Remote Live. Whether you enable Local Live or Remote Live staging, the interface for managing and publishing staged pages is the same.

So when should you use Local Live staging and when should you use Remote Live staging? Local Live staging lets you publish site changes very quickly, since the staged and live environments are on the same server. It's also easier to switch between the staged and live environments using Local Live staging. Since the staged content, however, is stored in the same database as the production content, your server needs to have more resources, and the content isn't as well protected or backed up as with Remote Live staging. Also, you can't install new versions of apps for testing purposes in a Local Live staging environment, since only one version of an app can be installed at any given time on a single Liferay server.

With Remote Live staging, your staging and live environments are hosted on separate servers, so your data is separated. This lets you deploy new versions of apps and content to your staging environment without interfering with your live environment. However, publishing is slower with Remote Live staging since data must be transferred over a network. Of course, you also need more hardware to run a separate staging server.

Visit the staging environment article (Local or Remote) that most closely aligns with your goal for staging content.

11.1 Enabling Local Live Staging

Staging allows changes to be made in a staging environment so that work can be reviewed, possibly using a workflow, before it's published to a live site. With Local Live staging, both your staging environment and your live environment are hosted on the same server. When Local Live staging is enabled for a site, a clone of the site is created containing copies of all of the site's existing pages. Typically, this means the staging and live environments share the same JVM, database, portlet data (depending on which portlets are selected when staging is enabled), and setup configurations, such as the properties set in the `portal-ext.properties` file. The cloned site becomes the staging environment and the original site becomes the live environment.

Site administrators can enable local staging for a site by navigating to the *Publishing* → Staging* menu. To get some hands-on experience with enabling Local Live staging, you can complete a brief example which creates a Local Live staging environment for your site.

1. Navigate to the Product Menu (left side) and select *Publishing* → *Staging*.

2. Select *Local Live*. You also have the option to enable page versioning and select staged content.

   You can enable page versioning on a site's public pages, private pages, both, or neither. Page versioning allows you to work in parallel on different versions of pages and maintains a history of all page modifications. You can also choose content for the staging environment to manage on the Staging Configuration page. You can learn more about these options in the Enabling Page Versioning and Staged Content article.

3. Click *Save*.

You've officially begun the staging process!

Because Local Live staging creates a clone of your site, it's recommended to only activate staging on new, clean sites. Having a few pages and some apps (like those of the example site you created) is no big deal. If you've already created a large amount of content, however, enabling staging can take a lot of time since it's a resource intensive operation. Also, if you intend to use page versioning to track the history of updates to your site, it's recommended that you enable it as early as possible, *before* your site has many pages and lots of content. Your site's update history won't be saved until you enable page versioning. Page versioning requires staging (either Local Live or Remote Live) to be enabled.

If you ever need to turn off the staging environment, return back to *Staging* from the Publishing dropdown. The processes you've created are displayed by default. Navigate to the *Options* icon ( ) from the upper right corner of the page and select *Staging Configuration*. Select the *None* radio button to turn Local Live staging off. Please note that this operation removes the staging environment altogether, so all content that was not published to your live site will be lost!

Great! Now you're ready to use Local Live Staging.

11.2 Enabling Remote Live Staging

When Remote Live staging is enabled for a site, a connection is established between the current site and another site on a remote Liferay server. The remote site becomes the live environment and the current site becomes the staging environment–an instance of Liferay used solely for staging. The remote (live) Liferay server and the local (staging) Liferay server should be completely separate systems. They should not, for example, share the same database. When Remote Live staging is enabled, all the necessary information is transferred over the network connecting the two servers. Content creators can use the staging server to make their changes while the live server handles the incoming user traffic. When changes to the site are ready to be published, they are pushed over the network to the remote live server.

When applying patches to a remote staging environment, you must apply patches to all of your servers being used. Having servers on different patch levels is not a good practice and can lead to import failures and data corruption. It is essential that all servers are updated to the same patch level to ensure remote staging works correctly.

Before a site administrator can enable Remote Live staging for a site, the remote Liferay server must be added to the current Liferay server's list of allowed servers. The current Liferay server must also be added to the remote Liferay server's list of allowed servers. You also need to specify an authentication key to be shared by your current and your remote server and enable each Liferay server's tunneling servlet authentication verifier. You can make all of these configurations in your Liferay servers' `portal-ext.properties` files. Your first step should be to add the following lines to your current Liferay server's `portal-ext.properties` file:

```
tunneling.servlet.shared.secret=[secret]
tunneling.servlet.shared.secret.hex=true
```

Then add the same lines to your remote Liferay server's `portal-ext.properties` file:

```
tunneling.servlet.shared.secret=[secret]
tunneling.servlet.shared.secret.hex=true
```

Liferay DXP's use of a pre-shared key between your staging and production environments helps secure the remote publication process. It also removes the need to send the publishing user's password to the remote server for web service authentication. Using a pre-shared key allows Liferay DXP to create an authorization context (permission checker) from the provided email address, screen name, or user ID *without* the user's password.

The values that you can specify for the `tunneling.servlet.shared.secret` property depend on the configured encryption algorithm, since different encryption algorithms support keys of different lengths. Please see the HTTP Tunneling properties documentation for more information. Note that the following key lengths are supported by the available encryption algorithms:

- AES: 128, 192, and 256 bit keys
- Blowfish: 32 - 448 bit keys
- DESede (Triple DES): 56, 112, or 168 bit keys (However, Liferay places an artificial limit on the minimum key length and does not support the 56 bit key length)

To prevent potential character encoding issues, you can use one of the following two strategies:

1. Use hexadecimal encoding (recommended). E.g., if your password was *abcdefghijklmnop*, you'd use the following settings in your `portal-ext.properties` file:

   ```
   tunneling.servlet.shared.secret=61626364656667686970616b6c6d6e6f70
   tunneling.servlet.shared.secret.hex=true
   ```

2. Use printable ASCII characters (less secure). This degrades the password entropy.

   If you don't use hexadecimal encoding, i.e. if you use the default setting `tunneling.servlet.shared.secret.hex=false`, the value of the `tunneling.servlet.shared.secret` property *must* be ASCII compliant.

Once you've chosen a key, make sure that value of your current server matches the value of your remote server.

Important: Do not share the key with any user. It is used exclusively for communication between staging and production environments. Any user with possession of the key can manage the production server, execute server-side Java code, or worse.

Next, you must allow the connection between the configured IPs of your app server and the Staging server. Add the following line to your remote Liferay server's `portal-ext.properties` file:

```
tunnel.servlet.hosts.allowed=127.0.0.1,SERVER_IP,[STAGING_IP]
```

The `[STAGING_IP]` value must be replaced by the staging server's IP addresses. If the server has multiple interfaces, each IP address must also be added, which would show as a source address for the http(s) requests coming from the staging server. The `SERVER_IP` constant can remain set for this property; it's automatically replaced by the Liferay server's IP addresses.

One last thing you'll need to do is update the *TunnelAuthVerfierConfiguration* of your remote Liferay instance. To do this, navigate to the Control Panel → *Configuration* → *System Settings* → *Foundation* → *Tunnel Auth Verifier*. Click */api/liferay/do* and insert the additional IP addresses you're using in the *Hosts allowed* field. Then select *Update*.

Alternatively, you can also write this configuration into an OSGi file (e.g., `osgi/configs/com.liferay.portal.security.auth default.config`) in your Liferay DXP instance:

```
enabled=true
hostsAllowed=127.0.0.1,SERVER_IP,[Local server IP address]
serviceAccessPolicyName=SYSTEM_USER_PASSWORD
urlsIncludes=/api/liferay/do
```

Remember to restart both Liferay servers after making these portal properties updates. After restarting, log back in to your local Liferay instance as a site administrator. Then navigate to the *Publishing* option in Site Administration and select *Staging*. Select *Remote Live* and additional options appear.

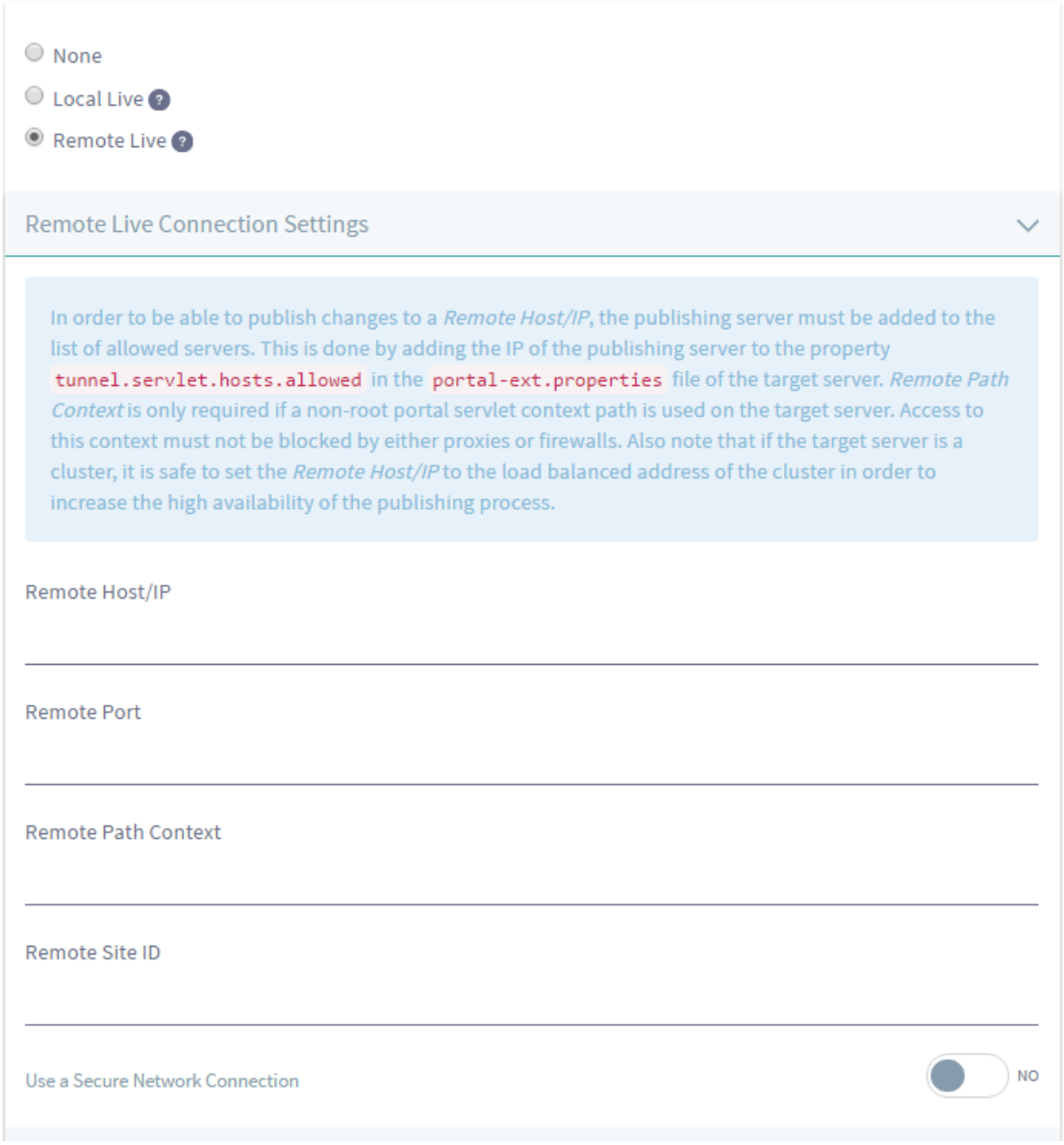


Figure 11.1: After your remote Liferay server and local Liferay server have been configured to communicate with each other, you have to specify a few Remote Live connection settings.

First, enter your remote Liferay server's IP address into the Remote Host/IP field. This field should match the host you specified as your `tunnel.servlet.hosts.allowed` property in the `portal-ext.properties` file. If you're configuring an IPv6 address, it must contain brackets when entered into the *Remote Host/IP* field (e.g., *[0:0:0:0:0:0:0:1]*).

Important: If you configured an IPv6 address, you must configure the app server's JVM to not force the usage of IPv4 addresses. For example, if you're using Tomcat, add the `-Djava.net.preferIPv4Stack=false` attribute in the `$TOMCAT_HOME\bin\setenv.[bat|sh]` file.

If the remote Liferay server is a cluster, you can set the Remote Host/IP to the load balanced IP address of the cluster to increase the availability of the publishing process.

Next, enter the port on which the remote Liferay instance is running into the Remote Port field. You only need to enter a Remote Path Context if a non-root portal servlet context is being used on the remote Liferay server. Finally, enter the site ID of the site on the remote Liferay server that will be used for the Live environment. If a site hasn't already been prepared for you on the remote Liferay server, you can log in to the remote Liferay server and create a new blank site. After the site has been created, note the site ID so you can enter it into the Remote Site ID field on your local Liferay server. You can find any site's ID by selecting the site's name on the Sites page of the Control Panel. Finally, it's best to check the *Use a Secure Network Connection* field to use HTTPS for the publication of pages from your local (staging) Liferay server to your remote (live) Liferay server.

Similar to Local Live staging, it is generally a good idea to turn remote staging on at the beginning of your site's development for good performance. When you're using Remote Live staging, and you are publishing a large amount of content, your publication could be slow and cause a large amount of network traffic. Liferay DXP's system is very fast for the amount of data being transferred over the network. This is because the data transfer is completed piecemeal, instead of one large data dump. You can control the size of data transactions by setting the following portal property in your `portal-ext.properties` file:

```
staging.remote.transfer.buffer.size
```

That's all you need to do to enable Remote Live Staging! Note that if you fail to set the tunneling servlet shared secret or the values of these properties on your current and remote servers don't match, you won't be able to enable staging and an error message appears. When a user attempts to publish changes from the local (staging) server to the remote (live) server, Liferay DXP passes the user's email address, screen name, or user ID to the remote server to perform a permission check. In order for a publishing operation to succeed, the operation must be performed by a user that has identical credentials and permissions on both the local (staging) and the remote (live) server. This is true regardless of whether the user attempts to publish the changes immediately or attempts to schedule the publication for later.

If only a few users should have permission to publish changes from staging to production, it's easy enough to create a few user accounts on the remote server that match a selected few on the local server. However, the more user accounts that you have to create, the more tedious this job becomes and the more likely you are to make a mistake. And you not only have to create identical user accounts, you also have to ensure that these users have identical permissions. For this reason, it's recommended that you use LDAP to copy selected user accounts from your local (staging) Liferay server to your remote (live) Liferay server. Liferay's Virtual LDAP Server application, available on Liferay Marketplace, makes this easy.

Remote Live Staging Verification

Before publishing *any* content, verify that **all** the necessary steps above have been completed. Otherwise, Remote Staging will fail.

1. The `tunnel.servlet.hosts.allowed` values are set in the `portal-ext.properties` file in both the staging and the production environment.

2. The `tunneling.servlet.shared.secret` values are set in the `portal-ext.properties` file in both the staging and the production environment.

3. The *TunnelAuthVerfierConfiguration* of your Liferay DXP instance is updated. This is a major change between legacy Portal and Liferay DXP. To do this:

    a) Navigate to the Control Panel → *Configuration* → *System Settings* → *Foundation*.
    b) Search for *Tunnel Auth Verifier*.
    c) Click */api/liferay/do* and insert the additional IP addresses you're using in the *Hosts allowed* field.
    d) Click *Update*.

4. The users who initiate the publishing processes exist on both the remote and staging environments. In addition, the users must have the same credentials (e.g., screen name, email, roles, and password).

You should proceed only when all four major steps have been completed.

11.3 Enabling Page Versioning and Staged Content

Enabling page versioning for a site lets site administrators work in parallel on multiple versions of the site's pages. Page versioning also maintains a history of all updates to the site from the time page versioning was enabled. Site administrators can revert to a previous version of the site at any time. This flexibility is very important in cases where a mistake is found and it's important to quickly publish a fix.

You can enable page versioning for public pages or private pages on the Staging Configuration page below the menu for selecting your staging environment (Local or Remote). If you've already enabled staging, you can navigate to the Product Menu → *Publishing* → *Staging* and click the () button and select *Staging Configuration*.

You can also choose content for the staging environment to manage on the Staging Configuration page.

Choosing content to be staged may sound self-explanatory, but content must have specific attributes in Liferay DXP to use it in a staged environment. Content or an entity should be site-scoped, so they are always part of a site; otherwise, they are not eligible for staging. Liferay DXP supports the following content groups for staging, by default:

- Application Display Templates
- Blogs
- Bookmarks
- Calendar
- Documents and Media
- Dynamic Data Lists
- Message Boards
- Mobile Device Families
- Polls
- Shopping
- Web Content
- Wiki

Before you activate staging, you can choose which of these applications' data you'd like to copy to staging. You'll learn about many of the collaboration apps listed under the Staged Portlets heading when you read the Collaboration Suite chapter. For now, you just need to be aware that you can enable or disable staging for any of these applications. Why might you want to enable staging for some application types but not others? In the case of collaborative apps, you probably *don't* want to enable staging since such applications are designed for user interaction. If their content were staged, you'd have to manually publish your site

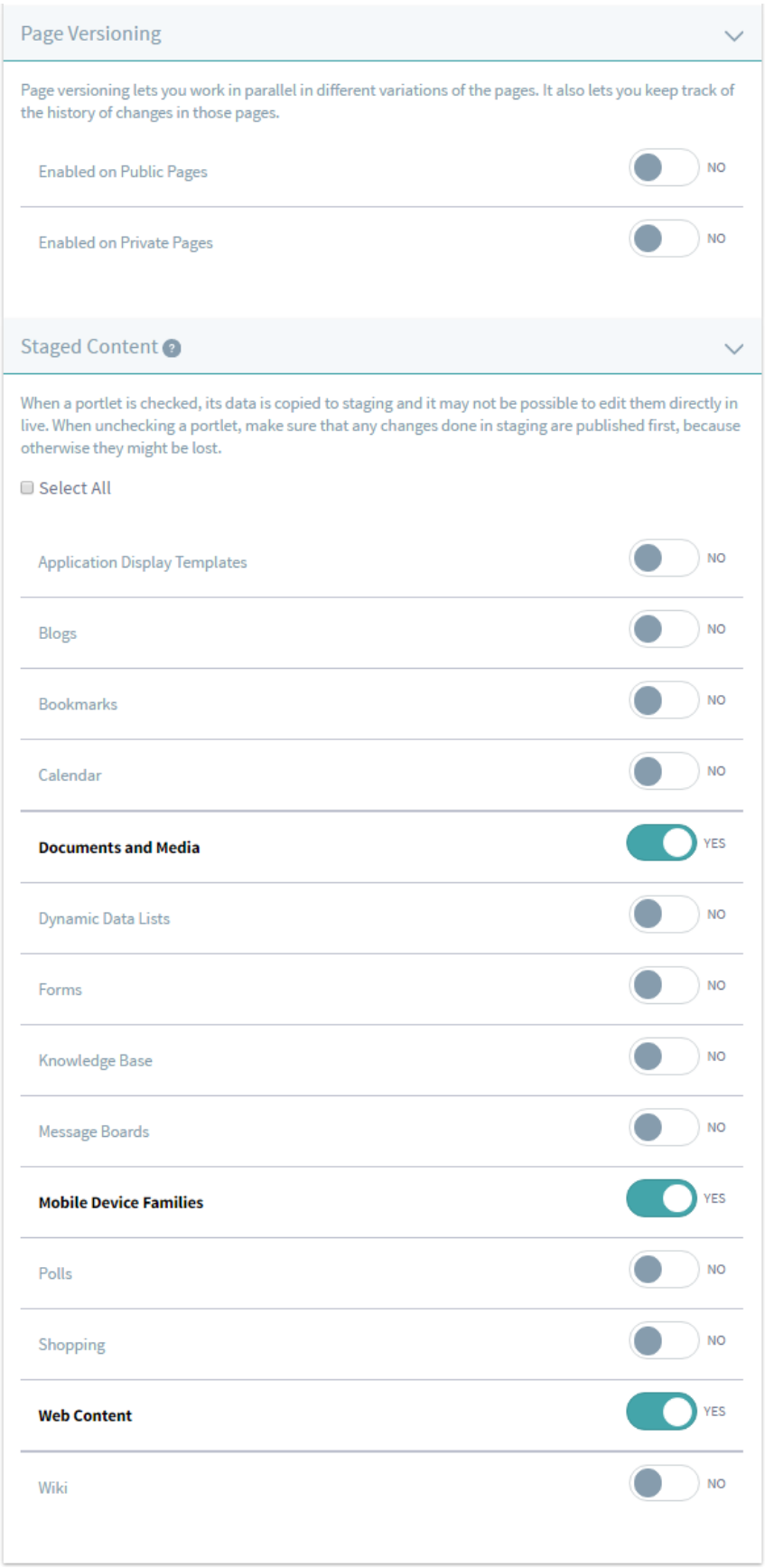


Figure 11.2: You can decide to use versioning and choose what content should be staged.

whenever somebody posted a message on the message boards to make that message appear on the live site. Generally, you'll want web content to be staged because end users aren't creating that kind of content–web content is the stuff you publish to your site. But applications like the Message Boards or Wiki would likely benefit from *not* being staged. Notice which applications are marked for staging by default: if you enable staging and accept the defaults, staging is *not* enabled for the collaborative apps.

The listed applications, or content groups, contain one or more specific entity. For example, selecting the Web Content application does not mean you're only selecting web content itself, but also web content folders.

Certain content types can be linked together and can reference each other on different levels. One of the responsibilities of staging is to discover and maintain these references when publishing. Site administrators and content creators have control over the process on different levels: staging can be enabled for a content group and a content group can be selected for publication.

Disabled staged content types can cause unintended problems if you're referring to them on a staged site. For example, the Asset Publisher portlet and its preferences are always staged. If the content types it's set to display are not enabled for staging, the Asset Publisher can't access them on a staged site. Make sure to plan for the content types you'll need in your staged site.

Turning Staging on and off for individual portlet data could cause data inconsistencies between the staging the live sites. Because of this, it's not possible to modify the individual portlet configuration once you enable staging. In case you need adjustments later on, you must turn staging off and re-enable it with your new configuration.

Besides managing the app-specific content, Liferay DXP also operates with several special content types such as pages or users. For instance, pages are a part of the site and can reference other content types, but in a special way. The page references apps, which means publishing a page also implies publishing its apps. The content gives the backbone of the site; however, content alone is useless. To display content to the end user, you'll need apps as the building blocks for your site.

Before you begin exploring the Staging UI, it's important to understand the publishing process for staging, and making informed decisions so you use the staging environment efficiently and effectively.

11.4 Publishing Staged Content Efficiently

Now that you have a firm grasp on how staging works, you'll dive deeper into the publication process and some prerequisites you should follow before publishing. By understanding how the process works, you can make smart and informed decisions about how you want to publish your site's content.

Understanding the Publication Process

In simple terms, publication is the process where all content, referenced content, apps and their preferences, pages, etc. are transferred from the staging scope to the live site. If you've enabled remote staging, this process involves network communication with another remote site. From a low level perspective, staging is an equivalence relation where entities are being mirrored to a different location. From a high level perspective, you can think of the staging publication process in three phases: export, validation, and import. These parts are executed sequentially.

During the export phase, the publication configuration is processed, which defines the site's contents and apps. This phase also gathers the obligatory referenced entities that will be required on the live site. Then everything according to the publication parameters has been processed into the instance's own file format, and that file has been stored locally or transferred to the remote live Liferay instance.

Next, the validation phase determines if it's possible to start the import process. This phase verifies the file's version and its integrity, checks for additional system information like language settings, and validates there is no missing content referenced.

Lastly, the import phase makes any necessary updates or additions to the site's content, layouts, and apps according to the publishing parameters. If everything is verified and correct, the staged content is published to your live site.

A crucial factor for successfully publishing staged content is data integrity. If anything is not successfully verified during the publication process, the transactional database can revert the site back to its original state, discarding the current publication. This is a necessary action to safeguard against publishing incomplete information, which could break an otherwise well-designed live site.

If the file system is not "database-stored" (e.g., DBStore), it's not transactional and won't be reverted automatically if a staging failure occurs. This could potentially cause a discrepancy between a file and where it's being referenced from. Because of this, administrators should take great care if they decide to stage the document library, making sure that regular backups of both their database and their file system are being maintained.

Next, you'll learn about staging best practices and prerequisites to follow for a seamless staging experience.

Planning Ahead for Staging

Staging is a complex subsystem of Liferay DXP that is designed to be flexible and scalable. Before advanced users and administrators begin using it for their site, it's important to plan ahead and remember a few tips for a seamless process. There are several factors to evaluate.

The most obvious factor is the content itself, including its amount, type, and structure. Depending on the content you'd like to use in your site, you can turn on staging for only the necessary content types, and leave others turned off to avoid unnecessary work. Publication can also be configured to publish only certain types of content.

The next factor to consider is the hardware environment. You should plan your environment according to the content types you're using. If your site operates on large images and video files, you should contemplate whether to use a shared network drive. For example, storing many large images in the Document Library usually requires a faster network or local storage. If you're dealing with web content, however, these are usually smaller and take up very little disk space.

The third major factor is possible customizations and custom logic for your staging environment. Your organization's business logic is most likely implemented in an app, and if you want to support staging for that app, you'll need to write some code to accomplish this. You can also consider changing default UI settings by writing new JSP code, if you want your staging environment's look and feel to change.

Once you've finished planning for your site, it is advised to turn on staging at the very beginning of the site creation process. This allows the site creator to avoid waiting for huge publications that can take long periods to execute. Taking smaller steps throughout the publication process forms an iterative creative process as the site is built from the ground up, where content creators can publish their changes immediately with no long wait times.

A few prerequisites to follow for staging are listed below:

- 4 GB of memory with 512 MB permgen
- 20 MB/s transfer rate minimum (disk)

Now that you know how the staging environment works and how to enable it for your site, you'll begin using it in the next section.

11.5 Using the Staging Environment

After enabling staging (either Local Live or Remote Live) for a site, you'll notice additional options provided on the top Control Menu and also in the Menu to the left. If you haven't enabled staging for your site, see the Enabling Staging section for instructions. These new menus help you manage staged pages. You'll also notice that most of your page management options have been removed, because now you can't directly edit live pages. You now must use the staging environment to make changes.

Staging Content

Click on the *Staging* button to view the staged area. Your management options are restored and you can access some new options related to staging.

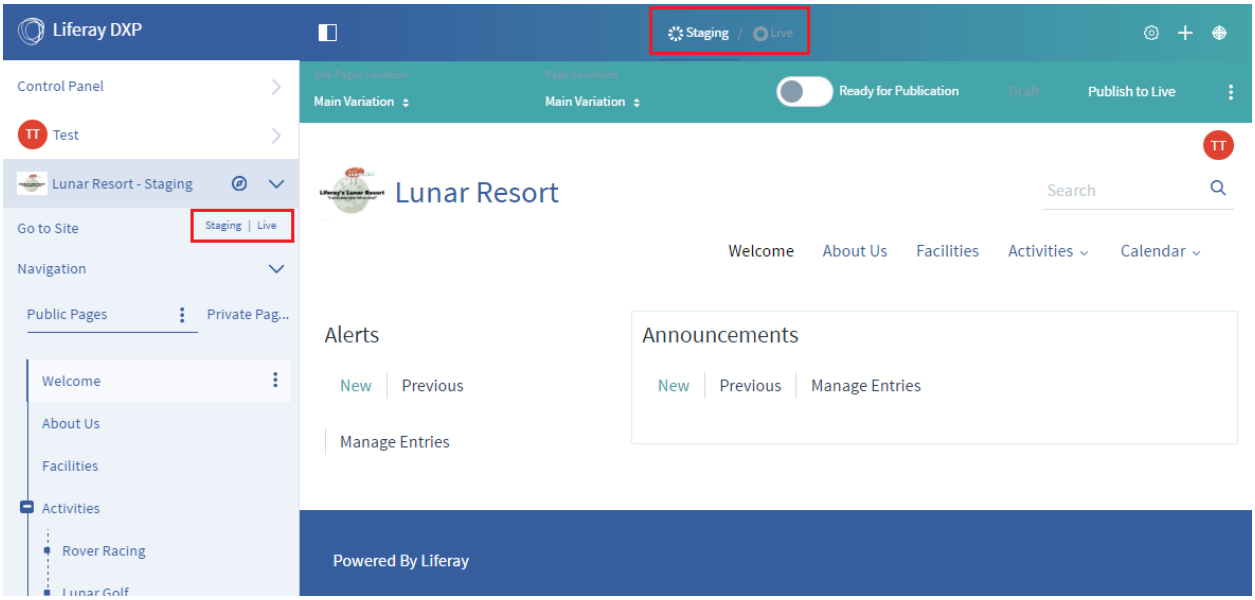


Figure 11.3: You can see the new staging options added to the top and left of your screen.

To test out the staging environment, add the Bookmarks application and then click on *Live* from the top menu. Notice that the Bookmarks app isn't there. That's because you've staged a change to the page but haven't published that change yet to the live site. Go back to the staged page and observe the options you have to choose from to help in your staging conquest.

Site Pages Variations: allows you to work in parallel on multiple versions of a staged site page. You'll learn more about this later.

Page Variations: allows you to work in parallel on multiple versions of a staged page. You'll learn more about this later.

Undo/Redo: allows you to step back/forward through recent changes to a page, which can save you the time of manually adding or removing apps if you make a mistake.

History: shows you the list of revisions of the page, based on publication dates. You can go to any change in the revision history and see how the pages looked at that point. To access *History*, select the *Options* icon () in the Staging bar.

Ready for Publication: After you're done making changes to the staged page, click this button. The status of the page changes from *Draft* to *Ready for Publication* and any changes you've made can be published to

the Live Site. When you publish a page to live, only the version which was *Marked as Ready for Publication* is published.

When clicking the *Publish to Live* button, a popup window appears with come configuration options for your publication. You can give your publication a name and view the changes since last publication. If everything looks good, you can click the *Publish to Live* button to publish your staged results to the live site.

If you'd like to further configure your publication, you can select the *Switch to Advanced Publication* button. Opening the Advanced Publication menu would be useful if you'd like to schedule a time to publish your content, edit the pages/content that will be included in the publication, manage permissions, etc. You're presented a custom publication menu, where you can perform advanced editing to your publication process.

You have two options to choose from for the Date category:

Now: immediately pushes any changes to the Live Site.

Schedule: lets you set a specific date to publish or to set up recurring publishing. You could use this, for example, to publish all changes made during the week every Monday morning without any further intervention.

Subsequently, you can choose additional options:

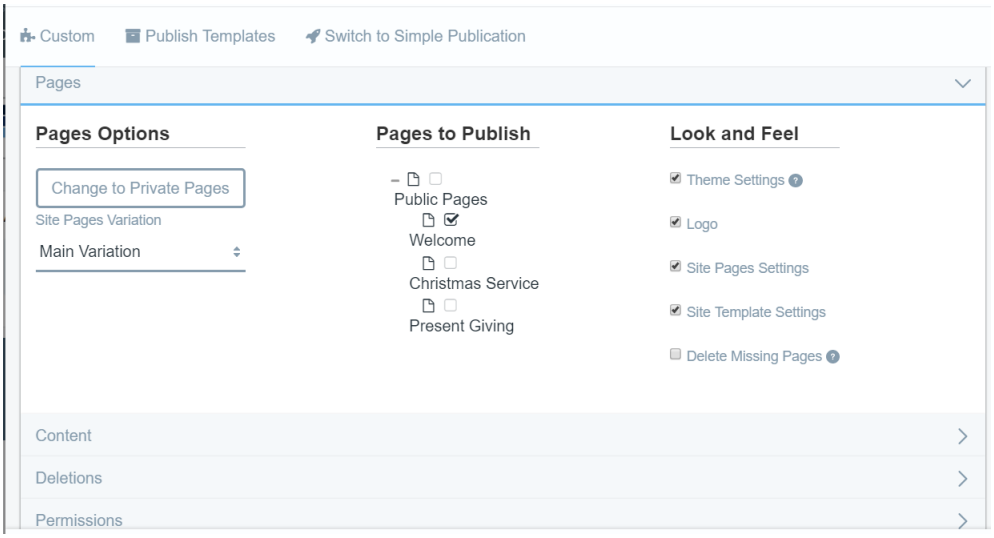


Figure 11.4: You're given additional publication options for your pages, content, deletions, and permissions.

Pages gives you the option to choose which pages to include when you publish. You can choose the page group (Public or Private) to publish by selecting the *Change to Public Pages* or *Change to Private Pages*. You cannot publish both at the same time; you'll have to complete their publication processes separately if you want to publish both page groups. You can also choose specific pages to publish, the look and feel of those pages, etc.

Note: If you're publishing pages with a custom theme, you must check the *Theme Settings* option under the Look and Feel heading for your staging configuration. If it's not checked, the default theme is always applied.

Content lets you configure the content to be published. Selecting the *Choose Content* option lets you filter the content to be published, based mainly on the date range and content type. If you choose a page to be published from the Pages menu, the portlets and their references are always published, even if you specified differently in the Content section.

There are other filtering sub-options for certain content types. Next, you'll look at these content filtering options.

You first must choose what content to publish based on date. Specifying a date range lets you choose content to publish based on when it was created or last modified. Select the option that best fits your workflow. The available options are described in more detail below:

All: publishes content regardless of its creation or last modification date.

From Last Publish Date: publishes content that was created or modified since the last publish date. This is the default option.

Date Range: publishes content based on a specified date range. You can set a start and end date/time window. The content created or modified within that window of time is published.

Last: publishes content based on a set amount of time since the current time. For example, you can set the date range to the past 48 hours, starting from the current time.

Under the date options are the different types of content that can be published. This list is populated based on the provided date range. For example, if at least one article has been created or modified in the given date range, a Web Content section appears in the list, and the number of articles is shown next to the Web Content label. Otherwise, the Web Content section is absent.

Categories and *Page Ratings* content types are not dependent on the date range, and are always shown in the list.

Note: Since some content types are meant for the end user and aren't supported in staging (e.g. comments, ratings, and custom fields), they can only be added to the live site and cannot be removed.

Unchecking the checkbox next to a certain content type excludes it from the current publication to Live.

Some of the content types in the list, like Web Content and Documents and Media, have further filtering options related to them. For instance, when the Web Content section is present and checked, it shows a comma-separated list of related items to be published, including the articles themselves. A sample list of related items for web content might look like this: *Web Content(12), Structures(3), Referenced Content, Version History*. You can remove items by clicking the *Change* button next to the list.

Referenced Content is represented by the Documents and Media files included in web content articles. Documents and Media content gets referenced when a user uses the editor to insert an image or if the article is based on a structure that has a field of the *Documents and Media* type.

Web content tends to be frequently updated, often more so than other kinds of content. Sometimes this can result in high numbers of versions, into the hundreds. This makes it take a long time to publish these articles. Liferay DXP addresses this issue by allowing you to choose whether or not to publish the *Version History*, or the past versions of the web content articles to be published. If you disable this option, only the last **approved** version of each web content article is published to Live. This can significantly speed up the publication process.

You can set this option globally. If you navigate to the Control Panel → *Configuration* → *System Settings* → *Web Content Administration*, you can toggle the *Publish version history by default* checkbox. This sets the default behavior. When publishing content, it is selected by default, so site administrators must manually uncheck the *Version History* box to publish only the latest approved version of web content articles. To change the default behavior, enable the checkbox in System Settings.

When the *Documents and Media* section is present (because at least one document has been created or modified in the provided date range), you can disable the transfer to live of the previews and thumbnails associated with the documents to be published. This can also speed up publication time.

Deletions lets you delete portlet metadata before publishing and delete operations performed for content types. If the *Replicate Individual Deletions* selector is enabled, operations performed for content in the staging environment are replicated to the target site.

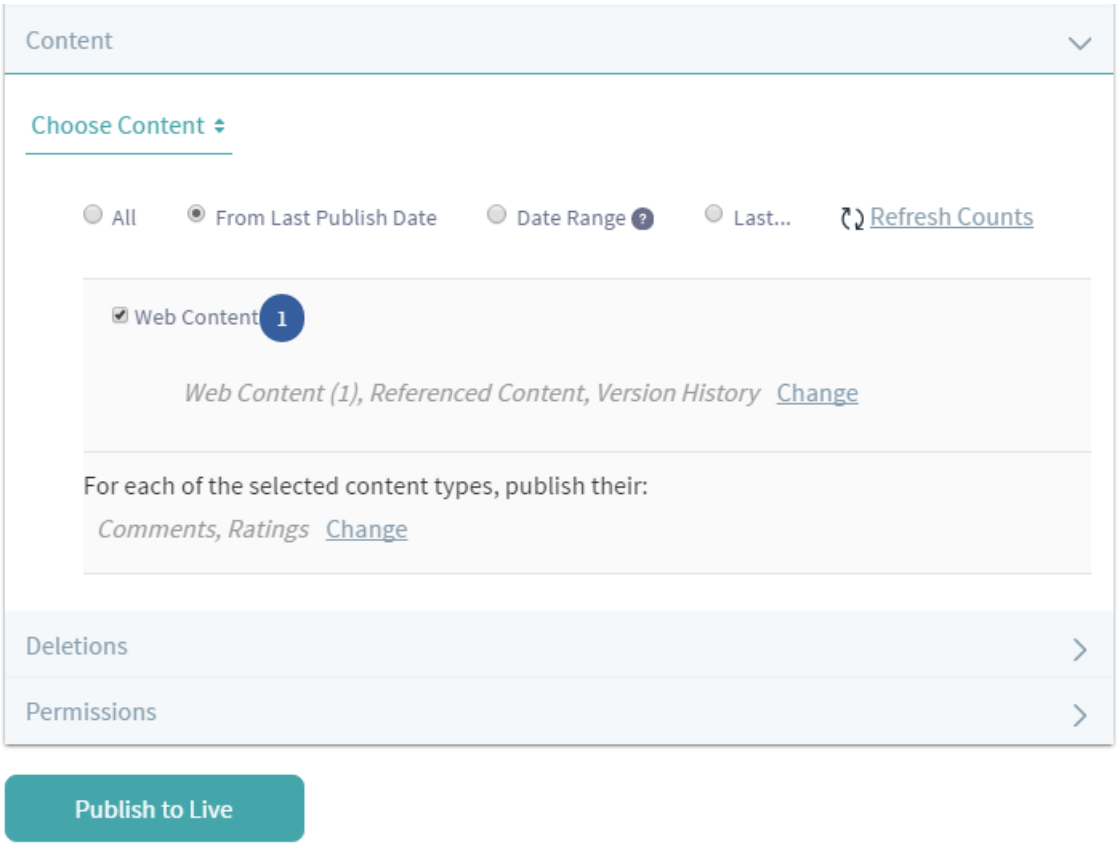


Figure 11.5: Click the *Change* button and uncheck the version history box to only publish the latest approved version of web content articles that have multiple versions.

Permissions allows you to include permissions for the pages and portlets when the changes are published.

Now that you have an idea of the available staging options, click *Mark as Ready for Publication*, and then click *Publish to Live* → *Now*. Select *Publish* to publish your Bookmarks application to the live site.

If you create a bookmark in the staged site, it isn't visible in the live site until you publish it to the live site. If workflow is enabled for any new resource, the resource needs to go through the workflow process before it can be published to the live site.

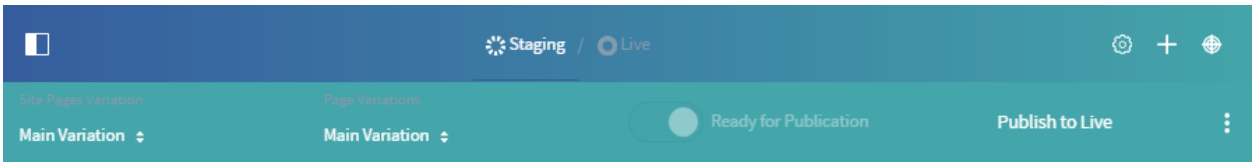


Figure 11.6: The staging toolbar indicates whether you're able to publish to the live site.

You can also manage your staging processes. Once staging is enabled, you can navigate back to the *Staging* option located in the Menu's Publishing tab. From here you'll see a list of staging processes that have been completed. You can relaunch or clear any of these publications by clicking the *Actions* button (⋮) next to a process. If you click the *Scheduled* tab from above, you'll find staging processes that you've scheduled for future publication dates.

If you find yourself repeatedly creating staging processes that are very similar to each other, you should

think about using Publish Templates. Up to this point, you've looked at how to create custom publication processes.

Instead of manually having to customize a publication process every time you're looking to publish pages/content, you can use a publish template. Using publish templates provides you the convenience of storing publication process settings so they can be reused. If you publish content frequently and usually select the same options to publish, the task of selecting options repeatedly can become tedious. With publish templates, you can select a custom template and immediately publish with the options you configured.

To create a publish template, select the *Options* icon ( ) from the top right corner of the Staging screen and select *Publish Templates*. Click the *Add* button ( ) and assign the template a name and description, and then fill out the configuration options as you would during a custom publication process. Once you've saved your publish template, it is available to use from the *Publish Templates* tab in the *Publish to Live* menu. To use the template, click the *Actions* button ( ) next to the template and select *Publish to Live*. This automatically fills the fields and options for publishing pages and their content. All you have to do is give the publication process a custom name. Once you confirm the configuration settings, your staging settings are published.

Note: When staging is enabled, the options available from your *Publishing Tools* tab are modified. When in the Live environment, you're only able to access the *Export* feature. When in the Staging environment, you're only able to access the *Import* and *Staging* features. The features that are not available for each environment don't make sense in that context. For example, you shouldn't be able to import content when in the live environment; it must be imported into the staged environment and then published before it is available in the live site.

Now that you know how to use the staging environment, you'll learn about the permissions involved to help manage this environment.

Managing Permissions

With the staging environment, there are many different options to use for building and managing a site and its pages. Sometimes limiting the access to a subset of the powerful features of staging is desired by some administrators. You can manage access to the staging environment by creating or modifying a role to possess certain permissions. To create/modify a role, navigate to the *Control Panel* → *Users* → *Roles*. You can create a new role by selecting the *Add* button ( ) and completing the New Role menu. Once you have a new role created, or you've decided on the role you'd like to modify, select the role's *Actions* icon ( ) and select *Define Permissions*.

The most obvious permissions for staging are the general permissions that look similar to the permissions for most Liferay apps. These permissions can be found in the *Site Administration* → *Publishing* → *Staging* section of the Define Permissions menu. This includes *Access in Control Panel*, *Configuration*, *Permissions*, *Preferences*, and *View*. Also, there are some site resource permissions that deal directly with staging. These permissions are located in the *Control Panel* → *Sites* → *Sites* section in the Define Permissions menu. The relevant site resource permissions related to staging are listed below:

- Add Page Variation: hides/shows the *Add* button on the Staging bar → Manage Page Variations screen.
- Add Site Pages Variation: hides/shows the *Add* button on the Staging bar → Manage Site Page Variations screen.
- Export/Import Application Info: if the Publish Staging permission is not granted, hides/shows the application level Export/Import menu. The Configuration permission for the Export/Import app is also required

- Export/Import Pages: if the Publish Staging permission is not granted, hides/shows the Export/Import app in the Site Administration menu.
- Manage Staging: hides/shows the staging configuration menu in the Staging app in the Site Administration menu.
- Publish Application Info: hides/shows the application level Staging menu.
- Publish Staging: hides/shows the *Publish to Live* button on the Staging bar and hides/shows the *Add* button in the Staging app in the Site Administration menu.
- View Staging: if Publish Staging, Manage Pages, Manage Staging, or Update permissions are not granted, hides/shows the Staging app in the Site Administration menu.

Notice that some of the permissions listed above are related to the export/import functionality. Since these permissions are directly affected by the Publish Staging permission, they are important to note. Visit the Importing/Exporting Pages and Content section for more details on importing/exporting site and page content.

One of the most powerful features of staging is page variations. Next, you'll see how to use them to create multiple different variations of your site's pages for different purposes.

11.6 Using Site Pages Variations

Let's say you're working on a product-oriented site where you'll have several major changes to a page or a set of pages over a short period of time. Also you need to be working on multiple versions of the site at the same time to ensure everything has been properly reviewed before it goes live. With staging, you can do this using *Page Variations*.

Notice that there are two page variation options available from the Staging bar: *Site Pages Variation* and *Page Variations*. Site Pages Variation is used to create different variations for the set of site pages. For instance, you could use this if you had three separate pages and wanted to modify these pages while keeping them together as a set. The Page Variations option only works with a single page.

For example, you can create several site pages variations, enabling the marketing team to give your site a completely different look and feel for Christmas. At the same time, the product management team can work on a different version that will be published the day after Christmas for the launching of a new product. Additionally, the product management team is considering two different ideas for the home page of the site, so they can create several page variations of the home page inside their product launch site.

Variations only affect pages and not the content, which means all the existing content in your staging site is shared by all your variations. In different site page variations, you can have different logos, different look and feel for your pages, different applications on these pages, different configuration of these applications and even different pages. One page can exist in just one site page variation or in several of them. Modifying the layout type (e.g., Layout, Panel, Embedded, etc.) or friendly URL of a page, however, **does** affect every site page variation.

You must enable page versioning when turning on Local or Remote Live staging. You can enable page versioning for public and private pages. When you turn staging on with page versioning enabled, the page variation options are available in the staging menu bar. By default, you only have one site page variation and page variation which are both called *Main Variation*. To create a new one, select the *Options* icon ( ) in the Staging bar and select the variation option. For example, select the *Site Pages Variation* option. This brings you to a list of the existing site page variations for your site. Click *Add Site Pages Variation* to create a new one.

From the *Add Site Pages Variation* screen, you can set a Name, Description, and also set your new variation to copy the content from an existing variation. There are several options to choose in this selector.

All Site Pages Variations: creates a new variation that contains the last version marked as ready for publication from any single page existing in any other variation.

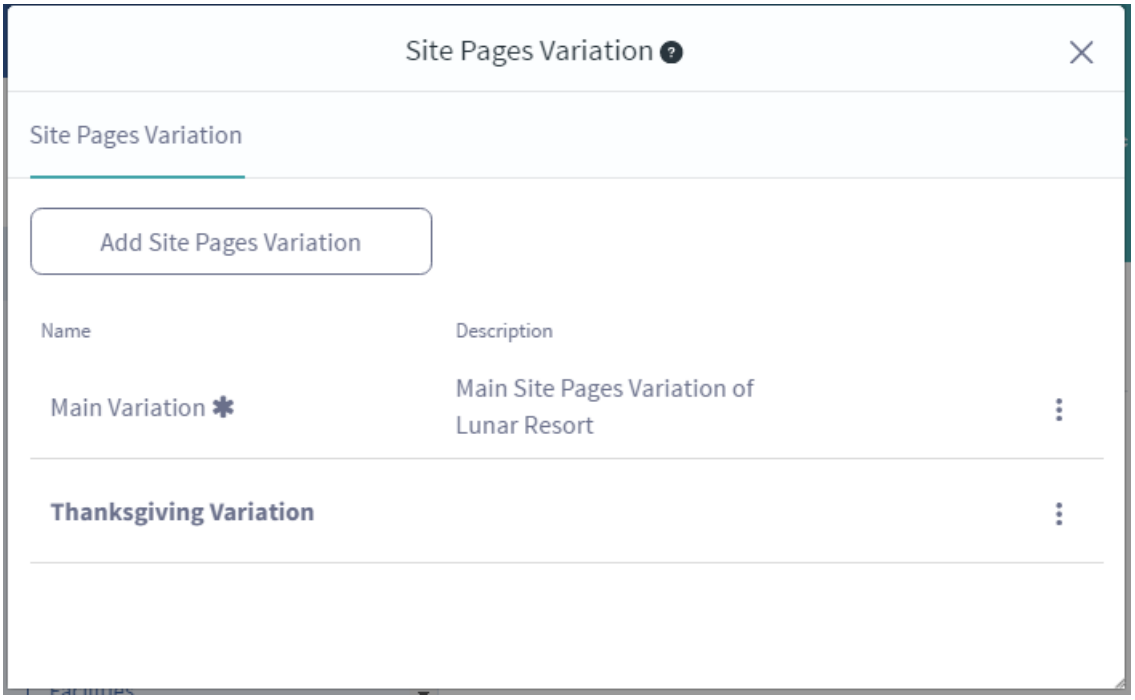


Figure 11.7: When selecting the *Site Pages Variation* link from the staging toolbar, you're able to add and manage your site pages variations.

None (Empty Site Pages Variation): creates a new, empty variation.

Main Variation: creates a new site page variation that contains only the last version of all the pages that exist in this variation. The current variation must be marked as ready for publication.

You are also able to rename any variation. For example, edit the Main Variation and change its name to something that makes more sense in your site, such as *Basic*, *Master*, or *Regular* and create a variation for.

You can switch between different variations by clicking on them from the staging menu bar. It's also possible to set permissions on each variation, so certain users have access to manage some, but not all variations.

You can now go to the home page of your Christmas variation and change the logo, apply a new theme, move applications around, change the order of the pages and configure different apps. The other variations won't be affected. You can even delete existing pages or add new ones (remember to *Mark as Ready for Publication* when you are finished with your changes).

To publish a variation to the live site, click on *Publish to Live* in the staging menu and then select *Publish to Live*. Publications can also be scheduled independently for different variations. For example, you could have a variation called *Mondays* which is published to the live site every Monday and another one called *Day 1* which is published to the live site every first day of each month.

You can also have variations for a single page inside a site page variation, which allows you to work in parallel in different versions of a page. For example, you might work on two different proposals for the design of the home page for the Christmas variation. These page variations only exist inside a site page variation.

To create a new page variation, click *Page Variations* on the staging toolbar. This brings you to a list of existing page variations for the current page (by default, there is only one called *Main Variation*). You can create more or rename the existing one. You can switch between different page variations using the dropdown menu on the staging toolbar. When you decide which page variation should be published, mark it as *Ready for Publication*. Only one page variation can be marked as ready for publication and that is the one that gets published to the live site.

For example, you could create a page variation called Thanksgiving for a page inside of the Christmas variation and another one called Christmas Day to display different content on those particular days.

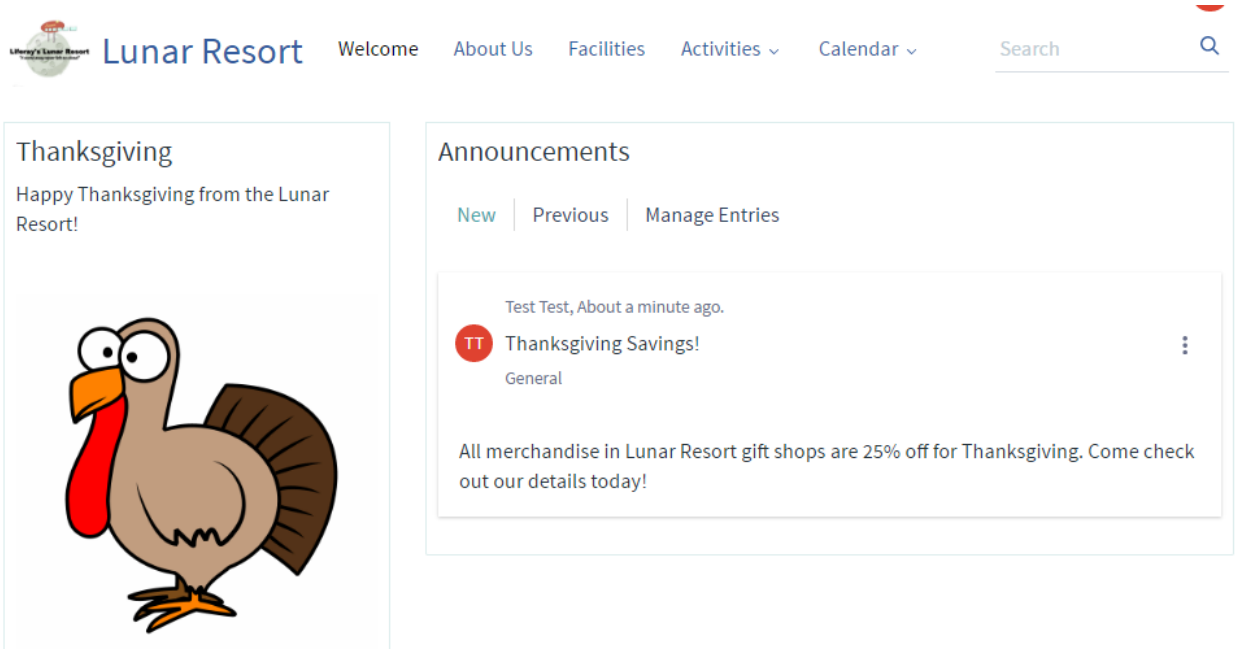


Figure 11.8: This is an example of a Thanksgiving page variation.

Another powerful feature is the possibility of *merging* Site Pages Variations. To merge two Site Pages Variations, you need to go to the Site Pages Variation screen. From there, click on *Merge* on the Site Pages Variation you want to use as the base. You will be asked to choose the Site Pages Variation to merge on top of it. Merging works in the following way:

- New pages that don't exist in the base Variation will be added.
- If a page exists in both Site Pages Variations, and at least one version of the page was marked as ready for publication, then the latest version marked as ready will be added as a new Page Variation in the target page of the base Variation. (Note that older versions or page variations not marked as ready for publication won't be copied. Merge can be executed, however, as many times as needed and will create the needed pages variations in the appropriate page of the base Site Pages Variation).
- Merging does not affect content nor will overwrite anything in the base Variation, it will just add more versions, pages and page variations as needed.

Liferay DXP's staging environment is extremely easy to use and makes maintaining a content-rich web site a snap. You'll learn about scheduling web content next.

11.7 Scheduling Web Content Publication

Liferay's WCM lets you define when your content goes live. You can determine when the content is displayed, expired and/or reviewed. This is an excellent way to keep your site current and free from outdated (and perhaps incorrect) information. The scheduler is built right into the form your users access to add web content. Specifically, it can be found in the bottom panel listed with several other configurable settings.

Display Date: Sets (within a minute) when content will be displayed.

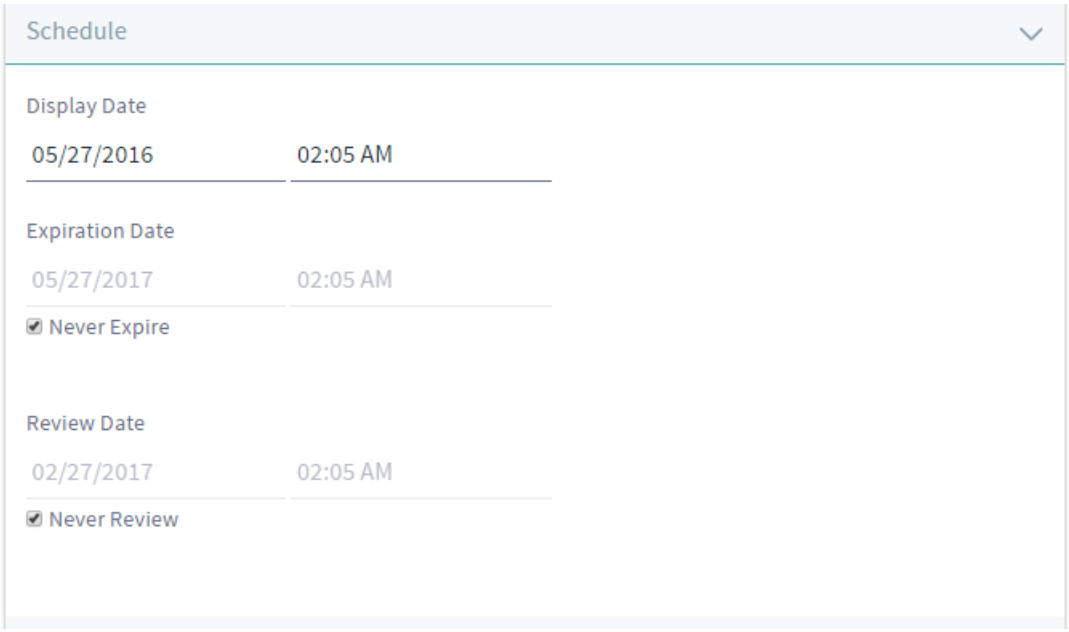


Figure 11.9: The web content scheduler can be easily accessed from the right panel of the page.

Expiration Date: Sets a date to expire the content. The default is one year.
Never Expire: Sets your content to never expire.
Review Date: Sets a content review date.
Never Review: Sets the content to never be reviewed.
As an example, you'll step through the process of scheduling a web content article.

1. Navigate to the Product Menu → *Content* → *Web Content*.
2. Create a new web content article by selecting the *Add Web Content* button ( ) → *Basic Web Content*.
3. Add content for your web content article.
4. Select the *Schedule* dropdown menu on the web content form. Configure the publication schedule.
5. Click *Publish*. Your web content article is now created and abides by the scheduling parameters you've set.

When you set a Display Date for an existing article it does not affect previous versions of the article. If a previous version is published, it remains the same until the new version is scheduled to display. However, the expiration date affects all versions of the article. Once an article has expired, no version of that article appears.

Tip: If you want only the latest version of articles to expire, and not every past version, go to *Control Panel* → *Configuration* → *System Settings* → *Web Content* and uncheck *Expire All Article Versions Enabled*. This makes the previously approved version of an article appear if the latest version expires.

The scheduling feature gives you great control in managing when, and for how long, your web content is displayed on your Site. Additionally, you can determine when your content should be reviewed for accuracy and/or relevance. This makes it possible to manage your growing inventory of content.

You can enable staging on an individual site basis, depending on your needs. This lets you put strict controls in place for your public web site, while opening things up for individual sites that don't need such strict controls. Another management feature used to control publishing of content is the web content scheduler. Similar to staging, the scheduler provides more control of your publishing process.

11.8 What's Expected from Staging?

Have you ever wondered what happens to an entity during the Staging process? Does Staging recognize the change? If so, how does it handle the update?

This article lists use cases for each Liferay entity. The use case tables have three columns:

- *Related entity:* the entity attached/related to the table's main entity. For example, in the Web Content Article (main entity) table, an attached entity could be a structure.
- *Action performed:* the type of modification completed on the related entity.
- *How does Staging handle this?:* how the Staging framework handles the action performed.

Navigate to the entity section you want.

Application Display Template (ADT)

The following table describes how entities characterized by an ADT are handled by the Staging framework.

The related entities that apply to this table are listed below:

- Asset Publisher
- Blogs
- Breadcrumb
- Categories Navigation
- Documents and Media
- Language Selector
- Navigation
- RSS Publisher
- Site Map
- Tags Navigation
- Wiki

Related entity	Action performed	How does Staging handle this?
Any entity from list above	ADT is added to entity	The ADT is published. ADTs and their related entities are handled separately by Staging.
Entity's ADT is modified	The ADT is changed and published.	
Entity's ADT is deleted	The ADT deletion is published, removing it from the live site.	

Related entity	Action performed	How does Staging handle this?
Existing ADT on live site is selected for entity	The entity portlet and its pages are published. If the page the entity resides on is published, its ADT is published too.	
ADT referenced from entity is updated	The ADT is marked as modified and published.	
ADT referenced from entity is deleted	The ADT is deleted on the live site and the entity falls back to the default template.	
Entity with ADT is removed from page	The ADT is unaffected and is removed with the entity when the page is published.	

Asset Resources

The following table describes how asset tags and asset categories are handled by the Staging framework.

Related entity	Action performed	How does Staging handle this?
Asset Category	Category moved into/out of vocabulary or another parent category	The category is marked as modified and is published.
Vocabulary/parent category holding categories is deleted	All categories in the vocabulary/parent category are deleted, removing them from the live site.	
Asset Tag	Two or more tags are merged into one tag	Merged tags are deleted, removing them from the live site.

Asset Publisher

The following table describes how entities that are displayed by the Asset Publisher (AP) are handled on the staged and live sites during the Staging process.

The related entities that apply to this table are listed below:

- Blog entry
- Bookmark
- Calendar event

- Calendar resource
- Knowledge Base article
- Documents and Media entities
  - Basic Document
  - Contract
  - Marketing Banner
  - Online Training
  - Sales Presentation

Related entity	Action performed	How does Staging handle this?
Any entity from list above	Entity added to staged site	The entity displays in staged AP, but not live AP.
Entity published to live site	The entity displays on both staged and live AP.	
Entity edited on staged site	The updated entity displays in staged AP, but not live AP.	
Updated entity published to live site	The updated entity displays on both staged and live AP.	
Entity deleted on the staged site	The entity does not display in staged AP, but does in live AP.	
Deleted entity published to live site	The deleted entity does not display on both staged and live AP.	
Dynamic asset selection	AP is published	The content according to the dynamic selection is published.
Dynamic selection is edited and AP is published again	The content according to the dynamic selection is published again.	
AP content is deleted	The content deletions are published by default, regardless of the AP.	
AP removed from page	Once the page is published, the AP is removed. Content is not affected.	
Dynamic selection exported is turned off	The content is not published with AP, but previously published content remains intact.	

Related entity	Action performed	How does Staging handle this?
Dynamic export limit is changed	The content specified in the dynamic export limit is published for display in AP.	
Manual asset selection	AP is published	The content according to the manual selection is published.
Manual selection is changed and AP is published again	The content according to the manual selection is published again.	
Manual selection content is deleted	The content according to the manual selection is deleted and not displayed in the AP.	
AP removed from page	Once the page is published, the AP is removed. Content is not affected.	
Manual selection exported is turned off	The AP does not publish any content.	
No limit for manual export	The AP publishes the exact same manually selected content.	

Audience Targeting

The sections below describe how Staging handles a specific Audience Targeting entity and the various actions that can be performed on its attached/related entities.

User Segment The following table describes how entities that are attached/related to an Audience Targeting user segment are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
User Segment	User segment is deleted	The user segment deletion is published.
Rule	User segment is published	All rules are published; if a rule refers to an entity created by the Web Content or Documents and Media portlets, it's published during their respective processes.
User segment is published with remote staging	The rule is published if the rule type is available in the destination environment; if not, the rule is not published.	

Related entity	Action performed	How does Staging handle this?
User segment is deleted	The user segment's rules are deleted and those deletions are published.	
Rule is added	The rule is published when its user segment is published.	
Rule is modified	The rule modification is published when its user segment is published.	
Rule is deleted	The rule deletion is published when its user segment is published.	
Asset category	User segment is published	The associated asset category is published.
User segment is deleted	The associated asset category is deleted.	
Report	User segment is deleted	The user segment's reports are deleted and those deletions are published.

Campaign The following table describes how entities that are attached/related to an Audience Targeting campaign are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
Campaign	Campaign is deleted	The campaign deletion is published.
User Segment	Campaign is published	The campaign's associated user segments are published.

Blogs

The following table describes how entities that are attached/related to Blogs are handled by the Staging framework.

Note: There are two types of images that can be used in Blog entries. They stem from two different frameworks: the legacy Image framework and the Document Library framework. The Document Library's `FileEntry` is the image entity created for any new Blog entries. The legacy Image framework's `Image` entity is provided for backwards compatibility for older Blog entries featuring images. Since there are no upgrade processes available for legacy Blogs, the old and new image entities are both supported. Therefore, they're both listed in the table below. Images that do not define the type can be assumed as the `FileEntry` type.

Related entity	Action performed	How does Staging handle this?
Blogs entry	Blog entry is created	The new entry is published.
Blog entry is exported	No changes to entry on site where it was exported from.	
Blog entry is exported without references	No changes to entry on site where it was exported from.	
Blog entry is imported	The entry and its references are available for publishing in the importation site. If any modifications were done on the entry before it's imported, the `modifiedDate` field is flagged in the resulting LAR, meaning when the entry is imported to a new site, it's only available in the staging site and not published by default to the live site.	
Blog entry is imported without references	The entry is available for publishing in the importation site. If any modifications were done on the entry before it's imported, the `modifiedDate` field is flagged in the resulting LAR, meaning when the entry is imported to a new site, it's only available in the staging site and not published by default to the live site.	
Blog entry is modified	The modified entry is published.	
Small image - (`FileEntry`)	Entry is deleted	The image is not deleted from the Blogs repo (e.g., Site Admin → *Content* → *Blogs* → *Images*).
Small image is deleted	The blog entry is not updated, but the image deletion is published.	
Small image is updated	The blog entry and image are updated and published.	

Related entity	Action performed	How does Staging handle this?
Blog entry is published with references	The small image is published.	
Blog entry is published without references	If the small image has already been published or is not included for publication, it is not (re)published.	
Small image - (`Image`)	Entry is deleted	The image is not deleted from the Blogs repo.
Small image is deleted	The blog entry is not updated, but the image deletion is published.	
Small image is updated	The blog entry and image are updated and published.	
Blog entry is published with references	The small image is published.	
Blog entry is published without references	If the small image has already been published or is not included for publication, it is not (re)published.	
Cover image	Add a blog entry with cover image before publishing to the live site and then delete the entry	The cover image is not deleted from the Blogs repo (e.g., Site Admin → *Content* → *Blogs* → *Images*).
Cover image is deleted in the Blogs repo	The blog entry still contains the cover image on the live site, but the image is unavailable in the Blogs repo.	
Cover image is updated	The blog entry and image are updated and published.	
Blog entry is published with references	The cover image is published.	
Blog entry is published without references	If the cover image has already been published or is not included for publication, it is not (re)published.	
Friendly URL	Custom URL is added	The new URL is published.
Custom URL is modified	The modified URL is published.	
Blog entry is deleted	The URL deletion is published.	

Related entity	Action performed	How does Staging handle this?
Blog entry is published	The URL is published when importing, adding, or updating an entry.	
Page link	Blog entry is published with references	The entry's references are added, validated, and published.
Page is deleted	The page deletion is published and page URL is broken in blog entry.	
Embedded image	Embedded image is added	The blog entry and image are published.
Embedded image is removed from blog entry	The blog entry is updated and published; the embedded image does not change.	
Embedded image is deleted	The blog entry is updated and published, and the image deletion is published.	
Blog entry is published with references	The embedded image is published.	
Blog entry is published without references	If the embedded image has already been published or is not included for publication, it is not (re)published.	
Embedded image - linking Liferay image with anchor	Embedded image is added with linking	The blog entry and embedded image are published.
Embedded image is removed from the blog entry	The blog entry is updated and published; the embedded image does not change.	
Embedded image is deleted	The blog entry is published with a broken link for the deleted embedded image.	
Blog entry is published with references	The embedded image is published.	
Blog entry is published without references	If the embedded image has already been published or is not included for publication, it is not (re)published.	
Expando (custom field)	Blog entry is deleted	The custom field value is deleted and published.
Blog entry is exported	The Expando values are exported with the columns.	

Related entity	Action performed	How does Staging handle this?
Blog entry is imported	The Expando values are imported. Columns are updated/added if they don't exist in the target system. If the Expando table does not exist, it's created.	
Custom field is modified	The entry is modified and published.	
Custom field's properties are modified	The entry is modified and published.	
Custom field is deleted	The entry is modified and published.	
Locales are modified	Locales are independent of a blog entry; they depend on portal languages. Therefore, they're not listed as modified for the blog entry.	
Deletions	Blog entry is deleted	The deletion is propagated and published.
Blog entry is added to Recycle Bin	The entry's removal is propagated and published.	
Blog entry is restored from Recycle Bin	The entry is published again.	
Permissions	Blog entry is deleted	The permissions are deleted and published.
Blog entry is exported	The permissions are exported.	
Blog entry is imported	The permissions are imported.	
Blog entry's permissions are modified	The entry is modified and published.	
Role is created	The entry is reindexed in the staging and live sites. This is required so the roles which view and entity are reindexed.	
Role is modified	The entry is reindexed in the staging and live sites. This is required so the roles which view and entity are reindexed.	

Related entity	Action performed	How does Staging handle this?
Role is deleted	The entry is reindexed in the staging and live sites. This is required so the roles which view and entity are reindexed.	
Team is created	The entry is reindexed in the staging and live sites. This is required so the roles which view and entity are reindexed.	
Team is modified	The entry is reindexed in the staging and live sites. This is required so the roles which view and entity are reindexed.	
Team is deleted	The entry is reindexed in the staging and live sites. This is required so the roles which view and entity are reindexed.	
Locales	Locales of the portal/site change	The article is imported, adding the value for the new default language from the previous default language.
Asset entry	File entry is exported	No changes.
File entry is imported	The asset entry is created automatically.	
File entry is deleted	The asset entry is deleted and the file deletion is published.	
Asset category	Category is added	The category is published.
Category is added to file entry	The file entry is updated and published.	
Category is updated	The updated category is published.	
Category is deleted	The category deletion is published.	
Category assigned to file entry is deleted	The file entry is updated and published.	
Category is removed from file entry, but not deleted	The file entry is updated and published.	
Asset tag	File entry is exported	The tags are exported.
File entry is imported	The tags are imported.	
Tag is added to file entry	The file entry is updated and published.	

Related entity	Action performed	How does Staging handle this?
Tag is removed from file entry	The file entry is updated and published.	
Asset link	New asset link is created in file entry	The file entry is updated and published.
Asset link is removed that links file entry and another entity	The file entry is updated and published.	
File entry is exported	The asset link is exported.	
File entry is imported	The asset link is imported if the target exists.	

Bookmarks

The following table describes how entities that are attached/related to Bookmarks are handled by the Staging framework.

Related entity	Action performed	How does Staging handle this?
Bookmarks entry	Entry is added to folder	The entry is published.
Entry is moved to a different folder	The entry is updated and published.	
Entry is deleted	The entry deletion is published; the folder is not affected.	
Folder is updated	The entry is not affected by the change.	
Folder is deleted	The folder and its entities are deleted.	
Bookmarks folder	Folder is added to another folder	The folder addition is published to live.
Child folder is moved to a different folder	The child folder change is published by default; the other folder is not affected.	
Parent folder is modified	The parent folder is modified and published. The children folders are not affected.	
Parent folder is deleted	The parent folder and its contents are deleted and those deletions are published.	

Related entity	Action performed	How does Staging handle this?
Child folder is modified	The child folder is published; the parent folder is not affected.	
Child folder is deleted	The folder deletion is published.	

Calendar

The sections below describe how Staging handles a specific Calendar entity and the various actions that can be performed on its attached/related entities.

Calendar Instance The following table describes how entities that are attached/related to a Calendar instance are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
Calendar resource	Resource is created	The calendar is modified and published.
Resource is modified	The calendar is modified and published.	
Resource is deleted	The calendar is modified and published.	
Tags	Tag for calendar is created	The tag is available on live site.
Related assets	Web content added to calendar as related asset	The web content's event link is available on live site.

Calendar Booking The following table describes how entities that are attached/related to a Calendar booking are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
Calendar	Event is added	The event displays on live site.
Event is added under new calendar	The event displays on live site.	
Calendar that contains booking is deleted	The booking is deleted.	
Calendar that contains booking is updated	The booking is not affected.	
Calendar booking is updated	The booking is published.	
	The calendar is unaffected by changes in its bookings.	

Related entity	Action performed	How does Staging handle this?
Calendar booking is deleted	The booking is deleted. The calendar is unaffected by changes in its bookings.	
Calendar booking is moved to different calendar	If the booking was in the staging site and moved to another site's calendar, then the live version of the original booking is deleted when published. If the booking was moved into a site that is in the staging site, then it's not visible until publishing.	
Staged site is invited to event	The event is added to the staging site and an admin can accept the invite.	
Non-staging site is invited to event	The event only displays on live site after changes have been published.	
Staged site invites user calendar to event	The user does not receive an invitation until the staging site publishes the event to the live site.	
Calendar booking	Calendar booking is deleted after the event has occurred	The event is removed from the live site.
Parent calendar booking	Parent calendar booking is deleted	The parent and its children are deleted once published.
Parent booking is changed	The parent booking is changed and published. The children are updated with the new changes. Their statuses should be preserved, except if the parent start/end time changes; in this case, children go back to pending. The reminder times of the children are never affected. If the parent or child is in the staging site, their changes should only propagate when it's published.	

Related entity	Action performed	How does Staging handle this?
Child booking is deleted	Child bookings cannot be deleted.	
Child booking is changed	The child booking is changed and published. The parent booking is unaffected.	
Child booking's parent is changed	The child booking is changed and published.	
Calendar resource	Calendar resource is updated	The Calendar event is published with the correct resource.
Calendar event is published after resources were added to the Calendar	The event only remains under the site's Calendar.	
Calendar resource is assigned to Calendar booking	If the resource is in the staging site, it will only appear on the booking once the resource's site is published. If the booking is in the staging site, the resource will only appear on the booking once the booking's site is published.	
Calendar resource is removed while it's assigned to Calendar booking	If the resource is in the staging site, the resource will be removed from the booking only when the resource's site is published. If the calendar booking is in the staging site, the resource will be removed from the live site immediately, even if the booking's site is not published. Otherwise, the resource is removed from the booking with no other changes.	
Calendar booking is deleted	This does not affect the Calendar resource.	
Calendar booking is updated	The booking changes are published, but the update does not affect the resource.	

Related entity	Action performed	How does Staging handle this?
Calendar resource is updated while assigned to Calendar booking	The resource is updated and published. The booking is not affected.	
Workflow	Page is published with Calendar event in a workflow	The event is pending on the live site until approval and new publication.
User is invited to Calendar event on staged site in a workflow	The user doesn't receive a Calendar event invite until approval and new publication.	

Calendar Notification Template The following table describes how entities that are attached/related to a Calendar notification template are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
Calendar	User published event with an invitation	The user receives the invitation.
User configures reminder email	The user receives the reminder.	
Page link	Invitation notification received	The user has the link to the Calendar event.
Reminder notification received	The user has the link to the Calendar event.	
Invitation template contains link to staged page	The template is published. The page URL must point to the live site's page during publication.	
Page link is removed from invitation template	The template is published with the removed page link.	
Referenced page is deleted from invitation template	The template is published, but possibly with a broken link.	
Referenced page is modified in invitation template	The page is published with no affect on the template.	
Reminder template contains link to staged page	The template is published. The page URL must point to the live site's page during publication.	

Related entity	Action performed	How does Staging handle this?
Page link is removed from reminder template	The template is published with the removed page link.	
Referenced page is deleted from reminder template	The template is published, but possibly with a broken link.	
Referenced page is modified in reminder template	The page is published with no affect on the template.	
Site link (not pointing to specific page)	Invitation notification received	The template body contains the correct site URL.
Reminder notification received	The template body contains the correct site URL.	
Site link added to invitation template	The template is published.	
Site link added to invitation template and invitation is received	The URL points to the site.	
Site link removed from invitation template	The template is updated and published; the site is not affected by the change.	
Site referenced in invitation's site link is deleted	The template is not affected by the change.	
Site referenced in invitation's site link is updated	The template is not affected by the change.	
Site link added to reminder template	The template is published.	
Site link removed from reminder template	The template is updated and published; the site is not affected by the change.	
Site referenced in reminder's site link is deleted	The template is not affected by the change.	
Site referenced in reminder's site link is updated	The template is not affected by the change.	
Embedded image	Embedded image added to invitation template	The image is published with the template.
Embedded image referenced in invitation template is deleted	The template is published even if the image is not available.	

Related entity	Action performed	How does Staging handle this?
Embedded image removed from invitation template	The template is published and the image is not affected.	
Invitation template containing embedded image is updated	The template is published and the image is not affected.	
Embedded image added to reminder template	The image is published with the template.	
Embedded image referenced in reminder template is deleted	The template is published even if the image is not available.	
Embedded image removed from reminder template	The template is published and the image is not affected.	
Reminder template containing embedded image is updated	The template is published and the image is not affected.	

DDL

The sections below describe how Staging handles a specific DDL entity and the various actions that can be performed on its attached/related entities. For more information on DDLs, see the Creating Data Lists article.

DDL Record The following table describes how entities that are attached/related to a DDL record are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
DDL record set	DDL record is added	The DDL record is published.
DDL record is edited	The modified DDL record is published.	
DDL record is deleted	The DDL record deletion is published.	
DDL record is submitted on live site	The DDL record does not appear on the staged site.	
DDL record is deleted on live site	The deleted DDL record is still present on the staged site.	
File entry from DDM form	User attaches file via Docs & Media field	The attached file cannot be viewed until the DDL list is published to the live site.
Page from DDM form	Display portlet is configured to display in Form view	The currently selected list does not affect the Form view until it's published to the live site.

Related entity	Action performed	How does Staging handle this?
Display portlet is configured to display in Spreadsheet view	The currently selected list does not affect the Spreadsheet view until it's published to the live site.	
Display portlet has no specific layout configurations	The currently selected list returns back to default view after it's published to the live site.	
New DDL list is selected in Display portlet	The new DDL list is not visible on live site until it's published.	
DDL list name is edited	The new name is available once it's published to the live site.	

DDL Record Version The following table describes how entities that are attached/related to a DDL record version are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
DDL record	DDL record version is deleted	A new version number is generated on the live site.
Workflow	DDL record is pending in workflow	The workflow is not present on the live site until it's approved an published.

DDL Record Set The following table describes how entities that are attached/related to a DDL record set are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
DDM structure	DDM structure is created	The new structure can only be viewed/edited on the staged site until it's published.
DDM structure is edited	The edit is not reflected in the DDL list on the live site until it's published.	
DDM template	DDL list is configured to have a form template	The form template and its field(s) and data definition field(s) are visible in any new DDL record after publishing them to the live site.

Related entity	Action performed	How does Staging handle this?
DDL Form template is removed	The default structure is displayed for DDL records after publishing the deletion to the live site.	
DDL list is configured to have a display template	The display template format is displayed for any new DDL record once it's published to the live site.	
DDL display template is removed	The default structure is displayed for DDL records after publishing the deletion to the live site.	
DDL record set (in Forms scope)	Form entry is submitted in staged site	The entry is not visible in the live site's Product Menu (*Content → Forms*) until it's published to the live site.
Form entry is submitted in live site	The entry is only visible in the live site's Product Menu (*Content → Forms*).	
Form entry is deleted in live site	The entry is not removed from the live site's Product Menu (*Content → Forms*) until the deletion is published to the live site.	
Form entry is edited from Forms Admin portlet	The edited form is no display on live site until it's published to the live site.	
Form entry is deleted from Forms Admin portlet	The deleted form is still present on the staged site.	

Forms The following table describes how entities that are attached/related to a DDL record set are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
Workflow	Form entry is submitted	The entry does not display on the live site until it's approved.

DDM Structures and Templates

The sections below describe how Staging handles DDM structures and templates and the various actions that can be performed on its attached/related entities. DDM structures and templates are typically used with web content. Visit the Web Content section for additional information on how Staging handles these entities when attached to web content.

DDM Structure The following table describes how entities that are attached/related to a DDM structure are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
DDM parent structure	Child structure is created	The parent structure is published with the new child structure.
Child structure is modified	The parent structure is published with the edited child structure.	
Child structure is deleted	The parent structure is published to account for the removed child structure.	
DDM data provider	Data provider is added	The data provider is not added to the live site.
Data provider referencing a form	The data provider is added to the live site.	
Data provider with form reference is modified	The data provider is modified on the live site.	
Data provider with form reference is deleted	The data provider remains on the live site.	
DDM form layout	New form is selected on staging site	The new form is visible on the live site.

DDM Template The following table describes how entities that are attached/related to a DDM template are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
DDM structure	DDM structure is edited and published to live via Admin portlet	The edited structure is available on the live site.
DDM structure is edited and published to live via site page	The edited structure is available on the live site.	
DDM template is added to a DDM structure	The template is available on the live site.	

Document Library

The section below describes how Staging handles a specific document library (DL) entity and the various actions that can be performed on its attached/related entities.

Related entity	Action performed	How does Staging handle this?
DL folder	Document moved into/out of a folder (workflow disabled)	The document is updated is published.
Document moved into/out of a folder (workflow enabled)	The document inherits the workflow configuration of its folder.	
Folder is modified	The modified folder is published.	
Folder is deleted	The folder and all files/folders contained within it are deleted.	
Folder restriction is changed	The folder and its entries are changed and published.	
Folder's permissions are modified	The folder is published.	
Folder is moved into/out of another folder (workflow disabled)	The folder and its file entries are updated and published.	
Folder is moved into/out of another folder (workflow enabled)	The folder and its file entries follow the configured workflow.	
Folder's workflow restrictions are changed	The folder is updated and published.	
Folder with workflow restriction is deleted	The folder deletion is published; the workflow is not affected by the folder deletion.	
Folder's workflow restriction is deactivated	The folder is updated and published; the folder falls back to using no workflow.	
DL file entry	File entry is added	The file entry is published.
File entry is deleted	The file entry deletion is published.	
File entry is updated (adding new file version)	The latest approved file version is published.	
Add a new type for file entry	The file entry is published.	
File entry is restored from Recycle Bin	The file entry is published.	

Related entity	Action performed	How does Staging handle this?
DL file version	Add new file version (editing the file)	The latest approved file version is published.
Delete latest approved file version	The deletion is published.	
Delete file version that is not the latest	No changes.	
DL file shortcut	File shortcut is added	The shortcut is published.
File shortcut is deleted	The deletion is published.	
File shortcut is restored from Recycle Bin	The shortcut is published.	
File shortcut's target file is changed	The shortcut on the live site redirects to the new target file.	
File shortcut's target file is deleted	The shortcut and target file deletions are published.	
DL file entry type	File entry type is added	The new file entry type is published.
File entry type is updated	The updated file entry type is published.	
File entry type is deleted	The file entry type deletion is published.	
Delete file entry type that belongs to file entry	Not allowed; file entry type cannot be deleted, so nothing is published.	
Update a file entry type assigned to a file entry	A new file version is created and published.	
Folder	External folder is updated	No changes.
Entities accessed through WebDAV	Staging folder is set up to be accessed with WebDAV	The folder is published.
New file is added to folder from the desktop through WebDAV	The file is added on the staging site and included in publication; however, it's not added immediately.	
File is deleted from folder on the desktop through WebDAV	The file is deleted on the staging site and the deletion is published to the live site.	
File is removed from the Documents & Media portlet through WebDAV	The file is deleted on the desktop and the deletion is published.	
Live folder is set up to be accessed with WebDAV	The user can access the folder's contents on the desktop.	

Related entity	Action performed	How does Staging handle this?
File is added to the live folder on the desktop	The file cannot be added since it has a write protection.	
Repository	Repository definition is added	The definition is published.
Folder or file entry is published	The repository's metadata is published.	
Repository metadata is updated in Liferay DXP	The repository definition is published.	
Repository definition is deleted	The deletion is published.	
Repository is exported/imported	All the repository's entries are exported/imported too.	
Repository entry	New repository entry is created	The entry is published when the repository definition is published.
Repository entry is deleted	The repository entry is immediately deleted even before publication.	
Expando (custom field)	File version is deleted	The custom field value is deleted.
Latest file version is exported	The custom field values are exported with the columns.	
File version is imported	The custom field values are imported. Columns are updated/added if they don't exist in the target system. If the custom field's table does not exist, it's created.	
Custom field value is modified	The new file version is created and published.	
Custom field value is modified for a Document Library folder	The folder is updated and published.	
Expando column is deleted	The file entry is updated and published.	
Custom field value is deleted	The file entry is published.	
Deletions	File entry is deleted	The file entry deletion is published.
File entry is moved to Recycle Bin	The file entry's removal is published.	
File entry is restored from Recycle Bin	The file entry is published again.	

Related entity	Action performed	How does Staging handle this?
Folder is deleted	The folder and its contents are deleted and those deletions are published.	
Folder is moved to Recycle Bin	The folder and its contents are removed and those removals are published.	
Folder is restored from Recycle Bin	The folder and its contents are restored and published.	
Comments	File entry is deleted	The file entry deletion is published.
File entry is exported	The comments are not exported with the file entry.	
File entry is imported	The comments are not imported with the file entry.	
Comment is modified	The modified comment is not published.	
Comment is deleted	The comment deletion is not published.	
Ratings	File entry is deleted	The file entry deletion is published.
File entry is exported	The ratings are not exported with the file entry.	
File entry is imported	The ratings are not imported with the file entry.	
Rating is modified	The modified rating is not published.	
Permissions	File entry is deleted	The permissions are deleted and published.
File entry is exported	The permissions are exported.	
File entry is imported	The permissions are imported.	
File entry's permissions are modified	The file entry is marked as modified and published.	
Role is created	The file entry is reindexed in both the staging and live sites.	
Role is modified	The file entry is reindexed in both the staging and live sites.	
Role is deleted	The file entry is reindexed in both the staging and live sites.	

Related entity	Action performed	How does Staging handle this?
Team is created	The file entry is reindexed in both the staging and live sites.	
Team is modified	The file entry is reindexed in both the staging and live sites.	
Team is deleted	The file entry is reindexed in both the staging and live sites.	
Asset entry	File entry is exported	No changes.
File entry is imported	The asset entry is created automatically.	
File entry is deleted	The asset entry deletion is published.	
Asset category	Category is added	The category is published.
Category is added to file entry	The file entry is updated and published.	
Category is updated	The category is published.	
Category is deleted	The category deletion is published.	
File entry's category is deleted	The file entry is updated and published.	
Category is removed from file entry, but not deleted	The file entry is updated and published.	
Asset tag	File entry with tags is exported	The tags are exported with the file entry.
File entry with tags is imported	The tags are imported with the file entry.	
Tag is added to file entry	The file entry is updated and published.	
Tag is removed from file entry	The file entry is updated and published.	
Asset link	New asset link is created with file entry	The file entry is updated and published.
Asset link is removed between a file entry and another entity	The file entry is updated and published.	
File entry is exported	The asset link is exported.	
File entry is imported	The asset link is imported if the target is there; if not, it's skipped.	

Polls

The sections below describe how Staging handles a specific polls entity and the various actions that can be performed on its attached/related entities.

Polls Choice The following table describes how entities that are attached/related to a Polls choice are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
Polls question	New question is created with choices	The question and accompanying choices are published.
Question is updated	The question is marked as modified and published.	
Choice is updated	The question is marked as modified and published with the updated choice.	
New choice is added to question	The question is marked as modified and published with the new choice.	
Question is deleted	The question and its accompanying choices are deleted and published.	

Polls Vote The following table describes how entities that are attached/related to a Polls vote are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
Polls choice	Vote is added to the choice	The vote is published.
Choice is updated	The question and accompanying choice are updated and published.	
Polls question	Question is deleted	The question and its accompanying choices and votes are deleted and published.
Question is updated	The question is modified and published; the accompanying votes are not affected.	

Site Administration

The following table describes how entities that are hosted in Liferay DXP's Site Administration menu are handled by the Staging framework.

Related entity	Action performed	How does Staging handle this?
Parent page	Create child page	The child page is published.
Change parent of a child page	The parent is updated and published.	
Delete child page via simple publication	The child page's deletion is published.	
Delete child page via advanced publication	The child page's deletion is published when the *Replicate Individual Deletions* functionality is enabled.	
Move child page to different parent	The child page is moved and published.	
Create child page in new site variation and publish via simple publication	The child page exists on the live site.	
Publish changes from the main variation without changes via simple publication	The child page does not exist on the live site.	
Page friendly URL	Friendly URL is set	The friendly URL is published.
Friendly URL is modified on new site page variation	The friendly URL is modified in every site page variation and published.	
Page icon	Icon is uploaded	The icon is published with the page.
Icon is replaced by another image	The icon is modified on the live site.	
Icon is deleted	The icon is deleted on the live site.	
Icon is modified in a new site page variation	The icon is published since it's set in the new site page variation.	
Page is published from the main variation without modification on the page icon	The icon from the main variation is display on the live site.	
Link to page	Link is set to a page	The link is published.
Delete page that is linked	The page deletion is published; the link is removed	
Remove link	The link removal is published.	

Related entity	Action performed	How does Staging handle this?
Link page to a page that exists in a variation only	A page existing in a different variation is not offered in the *Select Page* option.	
Portlets	Portlet is added to page	The portlet is published.
Portlet is added to a page in a new site page variation	The portlet is published.	
Portlet is removed from the page	The portlet's removal is published.	
Portlet is removed from a page in a new site page variation	The portlet's removal is published.	
Theme (reference only)	Theme settings are modified on specific page	The theme reference is updated on live site.
Theme settings are modified in a site page variation on specific page	The theme reference is updated on live site.	
New theme is configured for specific page	The new theme configuration is published.	
Public page theme settings are modified	The theme reference is updated on live site.	
Private page theme settings are modified	The theme reference is updated on live site.	
Layout template	Layout template is changed for page	The change is published.
Logo (individual page)	New image is uploaded as logo	The logo modification is published.
Uploaded logo is deleted	The logo modification is published.	
Logo (page set)	New image is uploaded as logo for public pages	The logo modification is published.
Uploaded logo is deleted from public pages	The logo modification is published.	
New image is uploaded as logo for private pages	The logo modification is published.	
Uploaded logo is deleted from private pages	The logo modification is published.	
Show Site Name option is disabled for public pages	The site name is not published.	
Show Site Name option is disabled for private pages	The site name is not published.	

Related entity	Action performed	How does Staging handle this?
Page template	Public page is created based on a new page template containing a theme modification	The page template's theme modification is published.
JavaScript	JavaScript is set for public pages	The JavaScript configuration is published.
JavaScript is removed from public pages	The JavaScript removal is published.	
JavaScript is set for private pages	The JavaScript configuration is published.	
JavaScript is removed from private pages	The JavaScript removal is published.	
CSS	CSS is set	The CSS configuration is published.
CSS is removed	The CSS removal is published.	
Page set (referenced by site template)	Change theme of new site template	The public pages created in the site template have the updated theme configured.

Web Content

The sections below describe how Staging handles a specific web content entity and the various actions that can be performed on its attached/related entities.

Web Content Article The following table describes how entities that are attached/related to a web content (WC) article are handled during the Staging process.

Related entity	Action performed	How does Staging handle this?
Folder	WC article is moved into/out of folder	The article is marked as modified and is published.
WC structure	WC article is published	The structure (and its parent structure) and template are published.
Structure is modified	All articles using it are reindexed. The structure is published.	
Parent structure is modified	All articles using the structure or child structure are reindexed. The structure is published.	

Related entity	Action performed	How does Staging handle this?
Structure is deleted	Structures with articles using it or other structures using it can not be deleted. There could be an issue on import deletion that cannot happen because there are still dependencies using this in live.	
WC template	WC article is published	The template is published and the template reference is validated. It fails if the template can not be found in a parent site or the global site.
Template is modified	The template is marked to be published. WC articles are ignored.	
Template is deleted	The templates with articles using it can not be deleted.	
Linked structure is modified	The template is marked to be published. Template could break, so warning is issued.	
Image or image URL contained in WC	WC article is published	The image(s) are published with article.
WC article is deleted	The image(s) in article are deleted.	
Image is modified	The article is published.	
Image is deleted	The article is published.	
Embedded file contained in WC	WC article is published	The embedded file is published with article.
WC article is deleted	No changes to embedded file.	
File entry is modified	The file entry is published. The article is not modified or published.	
File entry is deleted	The article is published. Be aware that the missing file could create a link to a non-existing document, resulting in a broken link.	
File entry's permissions are modified	No changes to WC article.	
Page link contained in WC	WC article is published	The page is imported; publishing won't fail if page is missing but users will get a warning message for a potential broken link.
WC article is deleted	No changes.	

Related entity	Action performed	How does Staging handle this?
Page is deleted	The article is published. A message displays warning that there may be a broken link.	
Page's friendly URL is modified	No changes.	
Site link (not pointing to specific page) contained in WC	WC article is exported	If the page exists, export the reference to it.
WC article is imported	Validate that the page exists.	
WC article is deleted	No changes.	
Page is deleted	The Web Content feature handles this action behind-the-scenes.	
Page's friendly URL is modified	The Web Content feature handles this action behind-the-scenes.	
Virtual hosts	WC article is published	The site's virtual host is updated.
WC article is deleted	No changes.	
Custom field (Expando) contained in WC	WC article is deleted	The custom field value is deleted.
WC article is exported	The custom field values are exported with the columns.	
WC article is imported	The custom field values are imported. Columns are updated/added if they don't exist in the target system. If the Expando table does not exist, it is created.	
Custom field value is modified	The article is marked as modified and is published.	
Custom field properties are modified	The article is updated.	
Custom field column is deleted	The article is updated.	
Custom field locales are modified	The Custom field's locale is independent of the web content's locale; they depend on the portal languages. They shouldn't be validated when publishing.	

Related entity	Action performed	How does Staging handle this?
WC article deletion	WC article is deleted in Staging	The deletion is propagated and the article is deleted in the live site.
WC article is moved to trash in Staging	The deletion is propagated and the article is deleted in the live site.	
WC article is restored from trash	The article is published again.	
WC comments	WC article is deleted	The comments are deleted.
WC article is exported	The comments are exported.	
WC article is imported	The comments are imported.	
Comment is modified	The comment is published by default (without the article, unless the article has also been modified, in which case the comment is also propagated).	
Comment is deleted	The comment is published by default (without the article, unless the article has also been modified, in which case the comment is also propagated).	
WC ratings	WC article is deleted	The ratings are deleted.
WC article is exported	The ratings are exported.	
WC article is imported	The ratings are imported.	
Rating is modified	The rating is published by default (without the article, unless the article has also been modified, in which case the rating is also propagated).	
WC permissions	WC article is deleted	The permissions are deleted.
WC article is exported	The permissions are exported.	
WC article's permissions are modified	The article is marked as modified and is published.	
WC role is created/modified/deleted	The article is reindexed, both in Staging and on the live site (because we are indexing the roles which can view an entity).	

Related entity	Action performed	How does Staging handle this?
WC team is created/modified/deleted	The article is reindexed, both in Staging and on the live site (because we are indexing the roles which can view an entity).	
Locales	The locales of the portal/site change	The article is imported properly, adding the value for the new default language from the previous default language.
Asset entry	WC article is deleted	The asset entry is deleted.
WC article is exported	No changes.	
WC article is imported	The asset entry is created or updated.	
Asset categories	WC article is deleted	No changes.
WC article is exported	The categories are exported.	
WC article is imported	The categories are created or updated.	
Category is deleted	The article is reindexed, but not published.	
Category is modified or moved	The article is reindexed, but not published.	
Category's permissions are changed	The article is reindexed, but not published.	
Asset tags	WC article is deleted	No changes.
WC article is exported	The tags are exported.	
WC article is imported	The tags are created or updated.	
Tag is deleted	The deletion is published (article should be reindexed, but not published).	
Tag is merged with other tag	The merge is published (article should be reindexed, but not published).	
Tag is modified	The tag is published (article should be reindexed, but not published).	
Asset links (current site)	WC article is deleted	The assets are deleted.
WC article is exported	The link is exported, but missing references are not added to related content.	
WC article is imported	The asset links with existing assets are imported.	

Related entity	Action performed	How does Staging handle this?
Related asset is deleted	The asset is ignored.	
Related asset is modified	The modified asset is published.	
Link to other asset is deleted from the other asset	The other asset is published, and it updates the asset link on import.	
Link is deleted		
Asset links (parent or global site)	WC article is deleted	The assets are deleted.
WC article is exported	The link is exported, but missing references are not added to related content.	
WC article is imported	The asset links with existing assets are imported.	
Related asset is deleted	The asset is ignored.	
Related asset is modified	No changes.	
Link to other asset is deleted from the other asset	No changes.	
Link is deleted		

Web Content Folder The following table describes how entities that are attached/related to a web content (WC) folder are handled during the Staging process.

Related Entity	Action performed	How does Staging handle this?
Parent folder	Parent folder is modified	The articles residing in the folder are published.
WC structure	Structure residing in folder is deleted	The structure can not be deleted if it is being used by a folder.
WC folder	WC article moved into/out of folder during workflow	The article inherits the workflow configuration of its folder.
Folder is deleted	All articles and folders within are deleted too and those deletions are published.	
Folder is modified	The folder is published.	
Folder restriction is changed	The folder and its entries are changed and published.	
Folder's permissions are modified	The folder is published.	

Related Entity	Action performed	How does Staging handle this?
Folder is moved into/out of folder without workflow	The folder and its entries are changed and published.	
Folder is moved into/out of folder with workflow	The folder and its entries are workflow enabled.	
Folder's workflow restrictions are changed	The folder is updated and published.	
Folder with a workflow restriction is deleted	The workflow is not affected by the folder deletion; the deletion is included for publication.	
Folder's workflow restriction is deactivated	The folder should fall back to using no workflow. The folder is updated and included for publication.	

Web Content Feed The following table describes how entities that are attached/related to a web content (WC) feed are handled during the Staging process.

Related Entity	Action performed	How does Staging handle this?
WC structure	Feed is exported	The structures and their reliant templates are exported too. If the structure or template doesn't exist, they are not exported.
	Feed is imported	The structures and their reliant templates are added/updated.
	Structure is deleted	The structure is deleted.
WC template	Template is deleted	The feed can exist without the template. It is not validated.
Renderer template is deleted	The feed can exist without the template. It is not validated.	
Page	Page's friendly URL is modified	The link to the page still works.
Portlet is removed from page	The feed can live without a portlet ID; the portlet is removed upon publication.	
Display page	WC article with display pages is configured	The page, Asset Publisher portlet, and content are published.
Display page is deleted	The page and its content are deleted.	

Related Entity	Action performed	How does Staging handle this?
WC article is deleted	The article deletion is published; the display page is not affected.	
WC article with a display page is added but the page was not published earlier and it's not included in the publication	The page reference is validated and the publication fails when the page is not there. A message displays explaining this to the user.	

Wiki

The following table describes how entities that are attached/related to Wikis are handled by the Staging framework.

Related entity	Action performed	How does Staging handle this?
Attachment link	Add a link to an attachment of the same page to a wiki page	The attachment is published and the link in the wiki points to the new attachment.
Add a link to another wiki page of the same node to a wiki page	The wiki page is published, but the link is not.	
Parent wiki page	Add a wiki page and a child of that page	Both pages are published.
Parent page is deleted	The parent page and all its children are deleted.	
Child page is deleted	The child page deletion is published.	
Parent page is updated	The updated page is published.	
Child page is updated	The updated page is published.	
Wiki redirect page	Page is moved to a new title	The pages are updated and published; the redirect works on the live site.
Intermediate page is deleted	The page deletion is published; the new page is not affected.	
Target page is deleted	Both the intermediate and new page are deleted and those deletions are published.	
Site link in content	Add a link to a Liferay site page to wiki	The new wiki content and the site page referenced in the link are published.

Related entity	Action performed	How does Staging handle this?
Wiki node	New node is added	The node is published.
Node is updated	The node is published.	
Set of pages is imported into existing node	The pages are published.	
Node is deleted	The node and all its dependent pages are deleted and those deletions are published.	
Wiki page	Page with no child/parent relationships is copied	The new page is published.
Page containing a child is moved, changing its title and parent	The page is published. After publication, the page is located under a new parent and has a new name. The child points to the new page.	
Page with no child/parent relationships is deleted	The page deletion is published.	
Attachment is removed from page	The page is published. The attachment is removed from the page on the live site.	
Category is removed from page	The page is published. The category is removed from the page on the live site.	
Page format is modified	The page is published.	
Image (not yet published) is added to page's content	The page and its image are published.	

Chapter 12

Publishing Content Dynamically

Any type of content in Liferay DXP is considered an asset. In the Creating Web Content chapter, you examined Liferay DXP's most common type of asset: web content. Other types of assets include blog posts, wiki articles, message board posts, bookmarks, and documents. It's possible for developers to define custom asset types that utilize Liferay DXP's asset framework. Originally, the asset framework was created to provide a mechanism for adding tags to blog entries, wiki articles, and web content without reimplementing the same functionality multiple times. The asset framework has been greatly extended since then and it now supports tags, categories, vocabularies, comments, ratings, and asset relationships.

The Asset Publisher application is designed to display multiple assets. It has quite a few configuration options which you'll cover in this chapter. By default, abstracts (previews) of recently published assets are displayed by the Asset Publisher app and links to their full views are provided. You can configure the Asset Publisher app to display a table of assets, a list of asset titles, or the full content of assets. You can also configure the Asset Publisher to display only certain kinds of assets and you choose how many items to display in a list. The Asset Publisher app is very useful for displaying chosen types of content, for displaying recent content, and for allowing users to browse content by tags and categories. The Asset Publisher is designed to integrate with the Tags Navigation and Categories Navigation apps to allow this.

This chapter covers the following topics:

- Tagging and categorizing content
- Using targeted, single value, and multi-value vocabularies
- Adding relationships between assets
- Geolocating assets
- Using faceted search
- Publishing assets
- Publishing RSS feeds
- Restoring deleted assets

The first thing you'll learn about is tagging and categorizing content.

12.1 Organizing Content with Tags and Categories

Tags and categories are two important tools you can use to help organize information in Liferay DXP. These tools help users to easily find the content they're looking for through search or navigation. Tagging and

categorizing assets is easy. You can tag or categorize an asset at creation time or when editing an existing asset. If you click on the *Metadata* section of the form when creating or editing an asset, you'll find an interface for adding tags and categories. If no categories are available to be added to the asset (e.g., if no categories have been created), the *Select* option won't appear.

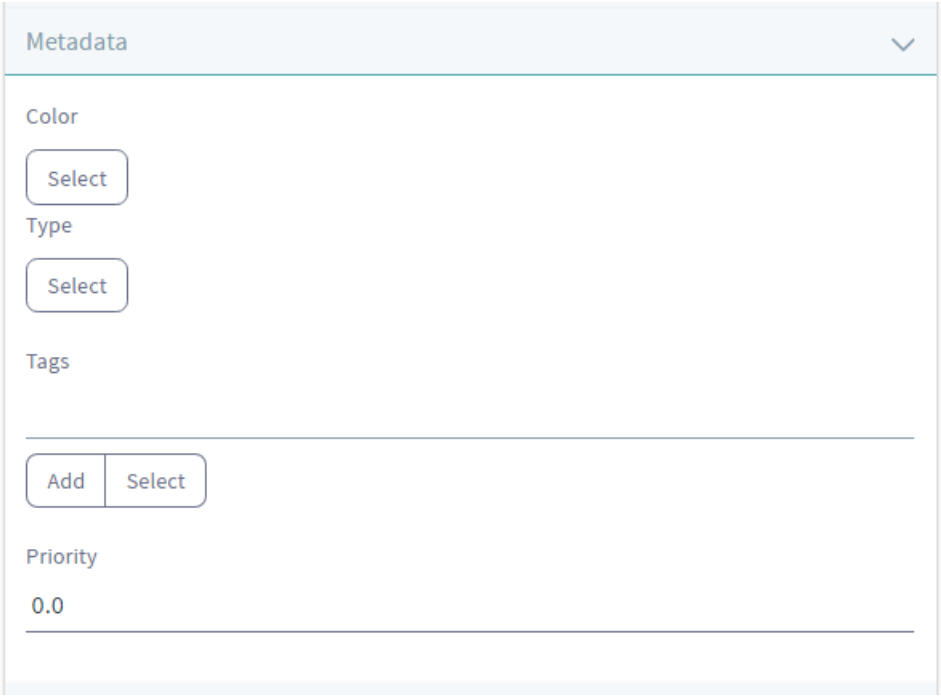


Figure 12.1: Here, the Web Content application's form for categorizing a new web content article includes the categories *Color* and *Type* with *Select* options, since categories exist.

Note: You'll notice in figure 1 above that there is also a *Priority* field for web content. This field is not related to categories and tags, but rather, specifies the order in which the web content article is listed when displayed in the Asset Publisher. To learn more about the Asset Publisher, see the Publishing Assets section.

The Menu ( ) contains interfaces for managing tags and categories for each site in Liferay DXP. Navigate to the Site Administration menu → *Content*, and you'll find the *Tags* and *Categories* options. These options can be used to manage all your site's tags and categories. It is important that you both tag and categorize your content when you enter it. You'll take a closer look at tags and categories next.

Tagging Content

Tags are an important tool that can help organize information in Liferay DXP and make it easier for users to find the content that they're interested in. Tags are lowercase words or phrases that you can attach to any content on the website. Uppercase characters in tags aren't recognized. Tagging content makes your search results more accurate and enables you to use tools like the Asset Publisher to display content in an organized fashion on a web page. There are two ways to create tags: you can do it through the administrative console in the Site Administration section of the Menu or on the fly as content is created. By default, tags can be created by regular users and users can apply them to any assets which they have permission to create or edit.

While regular users can, by default, create new tags by applying them to any assets that they have permission to create or edit, only site administrators can access the *Tags* application in the Content section

of the Site Administration area of the Menu. Here, site administrators can create new tags and edit any existing site tags. To create tags in Site Administration, visit the site for which you want to create tags and then click *Content* → *Tags*. From this screen, you can view existing tags and create new ones. To create a new tag, click the *Add Tag* icon ( ) and enter a name for the tag.

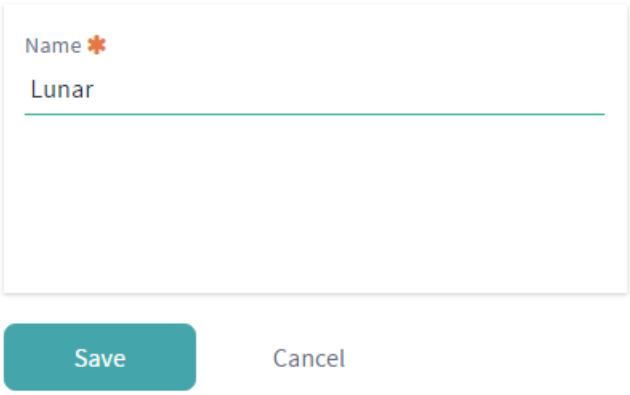


Figure 12.2: The Add Tag interface is very simple, only requiring the name of your tag.

The process for adding tags during content creation is very similar. For example, to create tags for a new web content article, navigate to the *Metadata* dropdown in a New Web Content menu, and add tags *Lunar*, *Moon*, and *Spectacular*. Once you've created the web content with these tags, the web content will be associated with those tag words when they are searched or referenced anywhere in Liferay DXP.

Tags are not the only instance-wide mechanism for describing content: you can also use categories.

Defining Categories for Content

Categories are similar in concept to tags, but are designed for use by administrators, not regular users. Hierarchies of categories can be created, and categories can be grouped together in *vocabularies*. While tags represent an ad hoc method for users to group content together, categories exist to allow administrators to organize content in a more official, hierarchical structure. You can think of tags like the index of a book and categories like its table of contents. Both serve the same purpose: to help users find the information they seek.

Note: In previous versions of Liferay, you could specify web content types via `portal.properties`. In Liferay DXP 7.0, web content types are no longer used and have been replaced by vocabularies. Vocabularies allow users to filter their web content articles by category instead, which lets you filter your content using the Asset Publisher and faceted search.

Adding vocabularies and categories is similar to adding tags. Visit the site for which you want to create categories and then click *Content* → *Categories* to view the Categories application.

Clicking on a vocabulary displays any categories that have been created under that vocabulary. To create a new vocabulary, click on the *Add Vocabulary* button ( ). Enter a name and, optionally, a description. By default, the *Allow Multiple Categories* option is enabled. This allows multiple categories from the vocabulary to be applied to an asset. If the box is disabled, only one category from the vocabulary can be applied to an asset. The *Associated Asset Types* lets you choose which asset types the categories of the vocabulary can be applied to and which asset types are *required* to have an associated asset from the vocabulary. Lastly, you can configure the permissions of the vocabulary. Should the vocabulary be viewable by guests? Only site

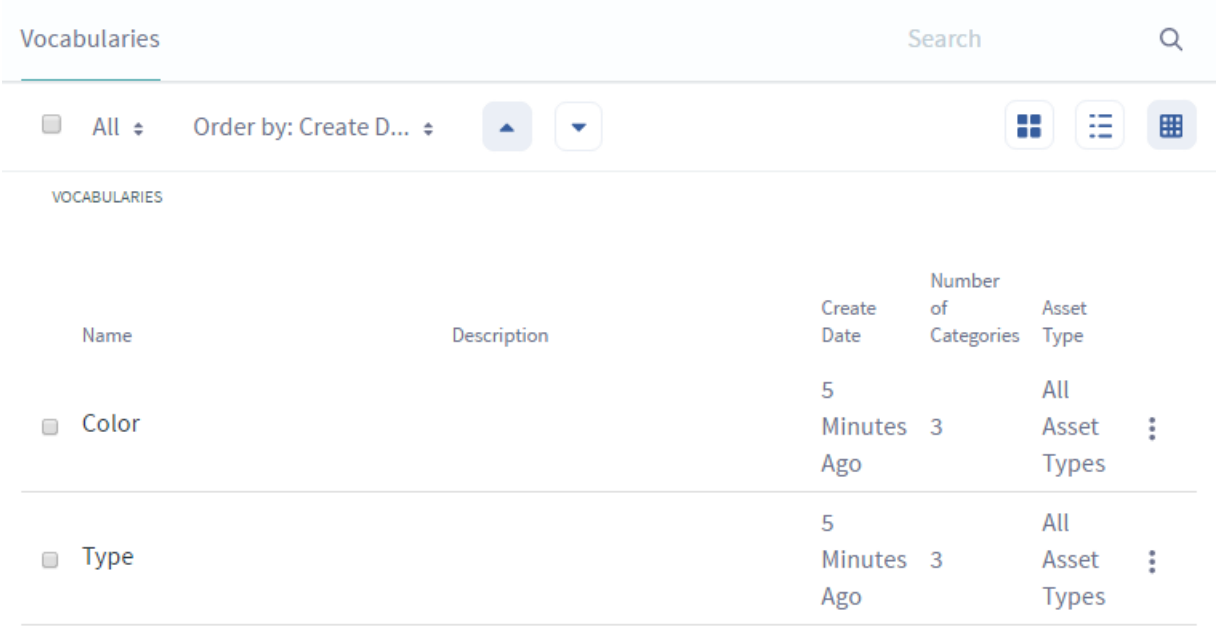


Figure 12.3: After adding new vocabularies, you'll notice your vocabularies indicate the amount of categories existing beneath them.

members? Only owners? Which of these roles should be able to delete the vocabulary, update it, or edit its permissions? By default, guests can view the vocabulary but only the owner can delete it, update it, or configure its permissions.

Creating new categories is similar to creating new tags except that categories must be added to an existing vocabulary and they can only be created by site administrators. Once created, however, regular users can apply categories to any assets they have permission to create or edit. To create a new category, click the *Add Category* icon ( ) if you're already viewing a vocabulary, or you can select the *Actions* button ( ) next to an existing vocabulary and select *Add Category*. Enter a name for the new category and, optionally, a description. Just as with tags, you can configure the permissions of the category, choosing which roles (guest, site member, owner) can view the category, apply it to an asset, delete it, update it, or configure its permissions. By default, categories are viewable by guests and site members can apply categories to assets. Also, you can add properties to categories. Category properties are a way to add information to specific categories. You can think of category properties as tags for your categories. Structurally, category properties are just like tag properties: they are key-value pairs associated with specific categories that provide information about the categories.

Once you have created some vocabularies and categories, you can take advantage of the full capabilities of categories by creating a nested hierarchy of categories. To nest categories, select the *Actions* button for the category you'd like to be the parent category. Then select *Add Subcategory*, which adds a child category to the selected parent.

After you've created a hierarchy of categories, your content creators will have them available to apply to content that they create. Click on *Web Content* in the Content section of Site Administration and click *Add* → *Basic Web Content*. Click on *Categorization* from the right-side menu and click *Select* on the vocabulary you'd like to apply. A dialog box appears with your categories. Select any relevant categories by checking the box next to them, and they'll be applied to the content.

Suppose you're running a Lunar Resort shop called Lunar Fireworks and you have many web content articles describing the colors and types of fireworks you offer. The abundance of your articles is overwhelming, and as your shop grows, so too does the web content articles you're required to manage. You've decided to

categorize your web content based on the color and type of firework, so the articles are easier to manage. Navigate to Site Administration → *Content* → *Categories* and create vocabularies *Type* and *Color*. Make sure both vocabularies are only used for web content articles by clicking the *Associated Asset Types* dropdown and selecting *Web Content Article*. Create categories *Fire* and *Smoke* for the Type vocabulary and *Red*, *Yellow*, and *Blue* categories for the Color vocabulary.

Now navigate to *Content* → *Web Content* in Site Administration and create an article called *Red Rocket*. This is your best selling product, so make sure to give it a detailed explanation and awesome picture. Select the *Metadata* dropdown for your web content article and select the Type → Fire and Color → Red categories. When you publish your new web content article for your best selling product, it'll be organized by its type and color. Once you've organized all your articles, you'll always be able to reference the type and color of a firework, just in case you forget.

There are a few other cool features for vocabularies and categories. A few of them were mentioned already when the *Allow Multiple Categories* and *Required* selectors for vocabularies and categories were discussed. The three new features are targeted vocabularies, single/multi-valued vocabularies, and separated widgets for every vocabulary.

Targeted Vocabularies Targeted Vocabularies allow you to decide which vocabularies can be applied to an asset type and which vocabularies are required for an asset type. To configure these settings, go to the Categories application in Site Administration and select a vocabulary's *Actions* icon. Select the *Associated Asset Types* tab to reveal a dialog box like the one below.

The default value for *Associated Asset Types* is *All Asset Types*. You can fine tune your choices by using the + and - buttons, which narrows the scope of the vocabulary to specific assets. In the screenshot above, notice that the vocabulary is configured to be available for Web Content articles and Blog entries, but it is not required. It is mandatory, however, for Bookmark entries.

Single and Multi-valued Vocabularies You can also decide if users can choose one or more categories from the same vocabulary to apply to an asset. If a vocabulary is single-valued you can only choose one. If it allows more, you can choose several categories from the vocabulary to apply to an asset.

You can configure the single-valued or multi-valued status of a vocabulary through the Categories application. Edit a vocabulary and deselect the *Allow Multiple Categories* selector to create a single-valued vocabulary. Use the default option to create a multi-valued vocabulary.

Separated Widgets A third feature of vocabularies and categories is that every vocabulary has its own separated widget. These widgets appear in the Categorization section of the form for editing an asset and they allow users to easily select appropriate categories for that asset.

It's important to use tags and categories with all your content, so that content is easier for users to find. Now that your content is categorized and tagged, you'll learn how to define content relationships next.

12.2 Defining Content Relationships

Related Assets is a feature that enables you to connect an asset to other assets within the same site or to global assets, even if they don't share any tags and aren't in the same category. The Related Assets application only displays content that meets the criteria you've specified, and also is listed as a related asset for a piece of content that is published using the Asset Publisher. You'll learn more about the Asset Publisher in the Publishing Assets section. For now, you'll focus on how to define relationships between assets so when you begin publishing assets, the Related Assets application can successfully display those relationships. You'll take a closer look at the Related Assets application next.

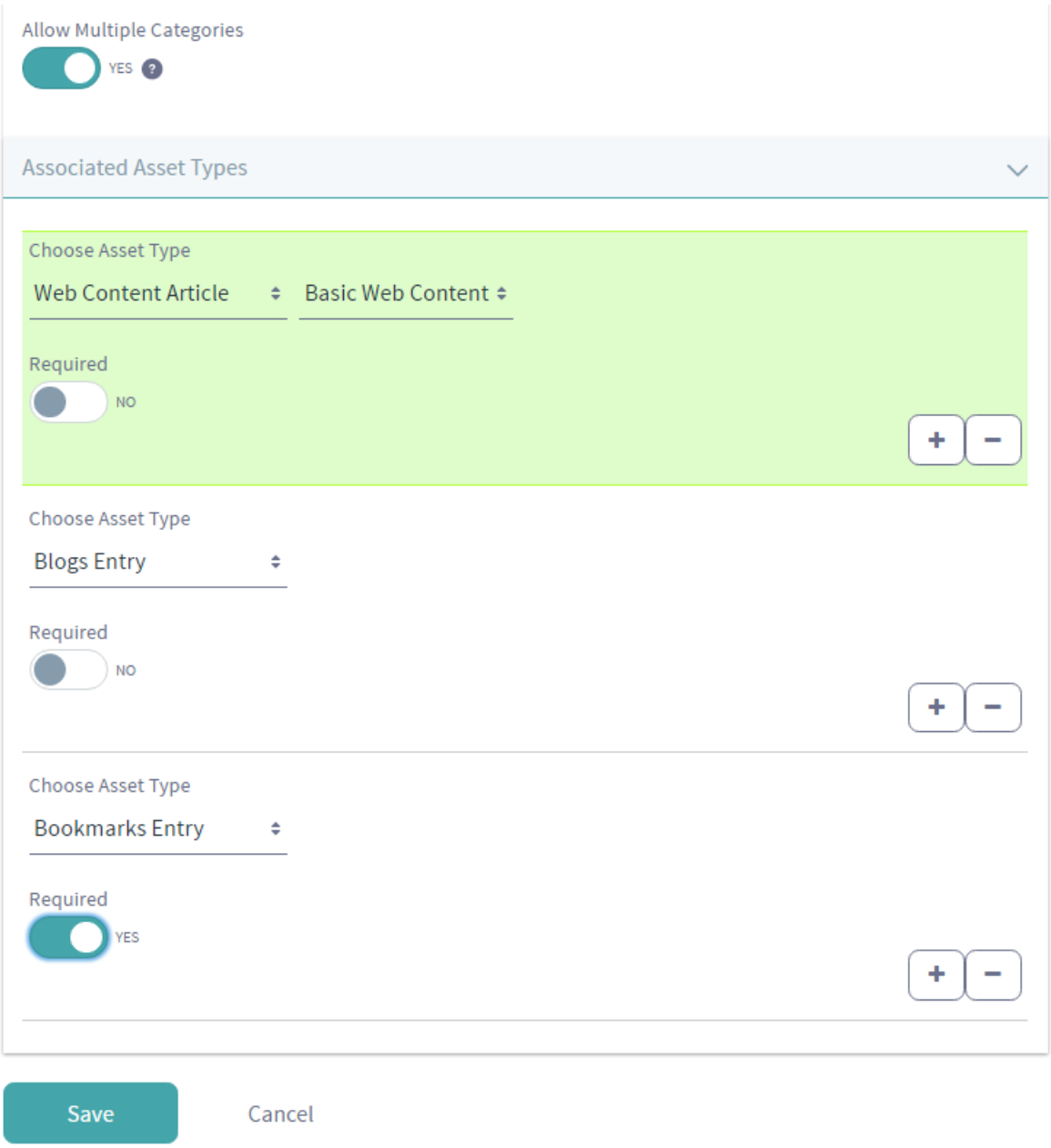


Figure 12.4: You can target vocabularies by checking the *Allow Multiple Categories* selector and then selecting the Asset Types.

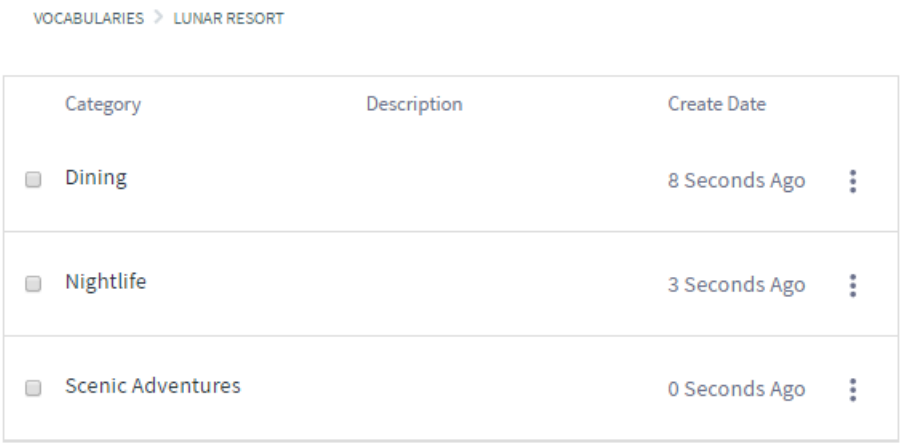


Figure 12.5: Multi-valued vocabularies allow multiple categories from the vocabulary to be applied to an asset. Single-valued vocabularies only allow one category from the vocabulary to be applied. Here, the *Dining* and *Nightlife* categories are selected to be applied but the *Scenic Adventures* category is not.

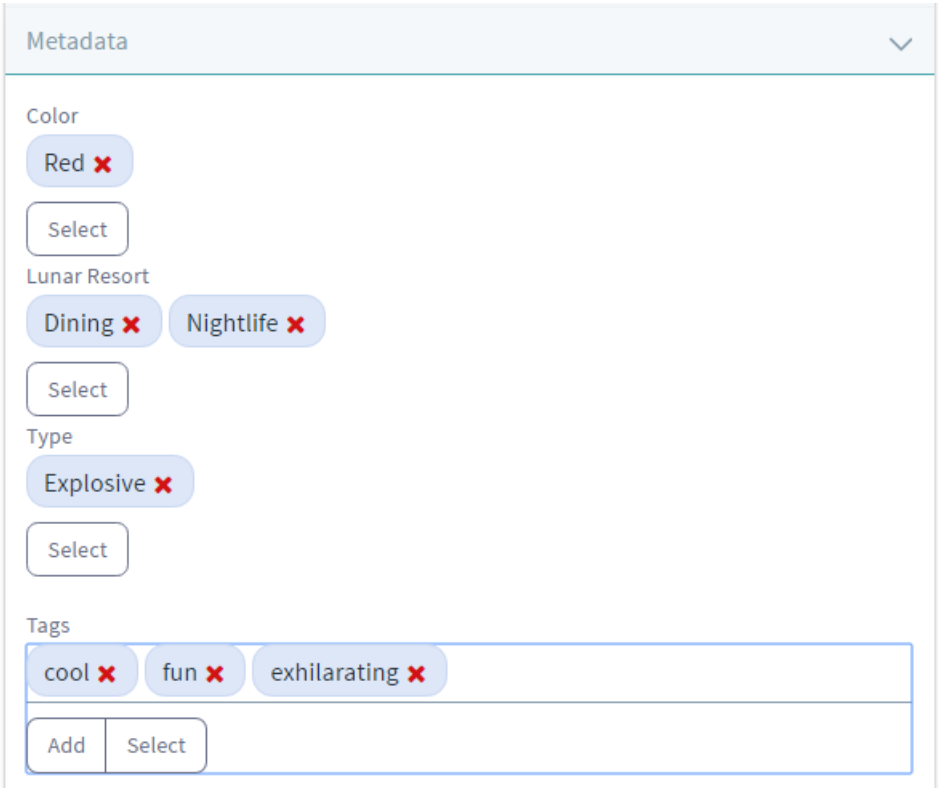


Figure 12.6: Vocabularies have their own widgets, making it easy to select available categories.

By default, the Related Assests application is configured to display any related asset of the asset selected in the Asset Publisher. You can configure what content relationships to display, if you prefer not to show every related asset for the selected asset. To do this, go to the Related Assets app and select the *Options* icon ( ) in the upper right corner of the application and click *Configuration*. Under the *Setup → Asset Selection* tab, set the type of asset(s) to display using the *Asset Type* menu. The default value is set to *Any*. You can narrow the scope of the app to display any single category of asset type or select multiple assets from the menu.

Filter options let you set minimum requirements for displaying assets by their categories, tags, and custom fields. Ordering and Grouping allows you to organize assets using the same criteria. Display settings allow you to customize how assets are shown in the app. They can be listed by title, in a table, by abstract, or full content. You can convert assets to different document types like ODT, PDF, and RTF. You can choose to show various metadata fields such as author, modification date, tags, and view count. You can even enable RSS subscriptions and customize their display settings. When you're finished setting the Source and Filter options, click *Save*.

Now that you've configured the Related Assets application to display specific content types, you need to actually define the relationships for your assets. Next, you'll go through a simple example of defining related assets for a web content article and then display those related assets.

Suppose you own a gift shop at the Lunar Resort, and you'd like all your shop's assets to display when an asset is clicked. You'll need to define relationships between your content, so when an asset is clicked, its related assets are displayed alongside the clicked asset. For this example, create a blog entry explaining your gift shop's new apparel and a photo of the moon, just so consumers are aware that you offer the *only* gift shop on a desolate rock orbiting the Earth!

Lastly, begin creating a web content article describing your shop. Once you've given your article a title and some content, open the *Related Assets* dropdown menu. Click *Select* and choose *Blogs Entry* and select the blog you created. Click *Select* again and choose *Basic Document* and select the photo of the moon. Click *Publish* to publish your web content article.

Now that those assets are created, you'll want to relate the blog entry and photo to your web content article. Navigate to your article in Site Administration → *Content* → *Web Content*.

You've now defined relationship with your three assets. Click the *Add* icon ( ) at the top of your page in the Control Menu and select *Applications* and add the Related Assets and Asset Publisher applications to the page. Wait a minute; there aren't any assets displayed in the related assets application. Why? You cannot see any related assets until you select an asset in the Asset Publisher.

Once you select an asset, its related assets are displayed in the Related Assets app, as in the image above. If you want to get even more detailed with how your related assets are displayed, you can place two Related Assets applications on the page and name one *Related Blogs* and the other *Related Photos*. To change the name of an application's title, click the application's *Options* icon and select *Look and Feel Configuration*. Select the *Use Custom Title* checkbox and provide the custom title.

Next, you'll learn how to use Liferay DXP's Geolocation feature in your assets.

12.3 Geolocating Assets

Geolocation is a feature that enables you to add the geographic coordinates where an asset was created, as metadata to your assets. You can add geolocation metadata to your web content, data lists, and documents & media. This feature is provided for you out-of-the-box by Liferay DXP 7. However, you must first enable it in your assets in order to use it.

Having the ability to Know where your user data is coming from, allows you to visualize patterns quickly and efficiently, so you can create a optomized experience that caters to the needs of the end user.

Next, you'll examine how you can enable geolocation in your web content.

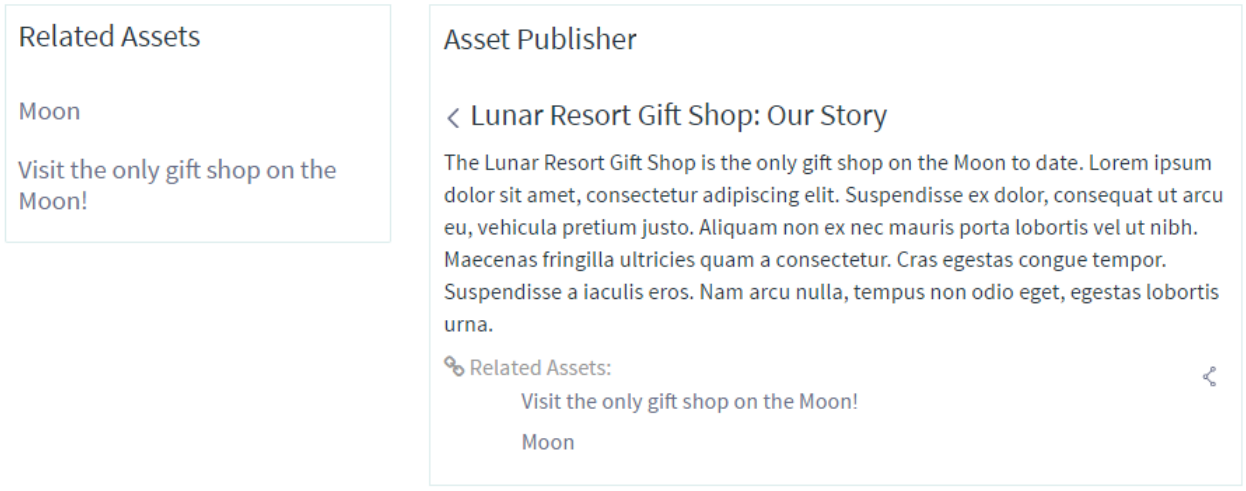


Figure 12.7: Select an asset in the Asset Publisher to see its related assets displayed in the Related Assets application.

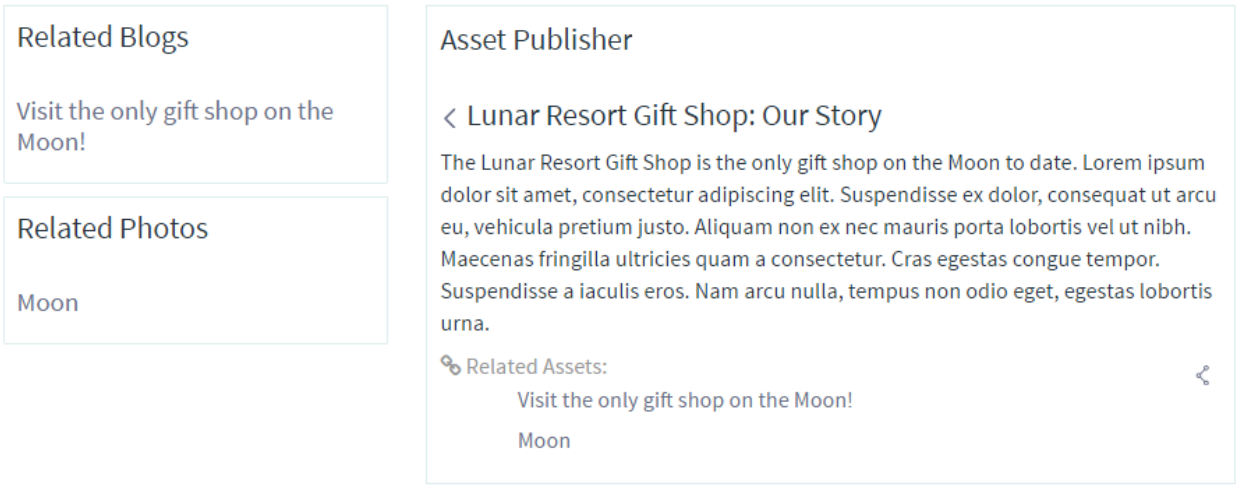


Figure 12.8: Related Assets applications can be configured to display specific content.

Geolocating Web Content

To use geolocation in your web content, make sure your structure includes a Geolocation field.

Once the structure is created, creating a template that uses the geolocation is a piece of cake. Create a new template and select the structure you just created with the geolocation field. Scroll down to the *Script* heading and locate the *Fields* section. Here you will see *Content* and *Geolocation* snippets. Click on the snippets to add them to the template, and *Save*.

To set your location for the web content, you can share your location with the browser, type a specific address into the address bar on the map, or even drag the indicator and drop it in any point in the map and the address will be automatically updated to reflect the new point. Once the web content is saved, the location is added as metadata to the web content.

Note: Depending on your browser settings, you may need to configure it, so the geolocation can obtain

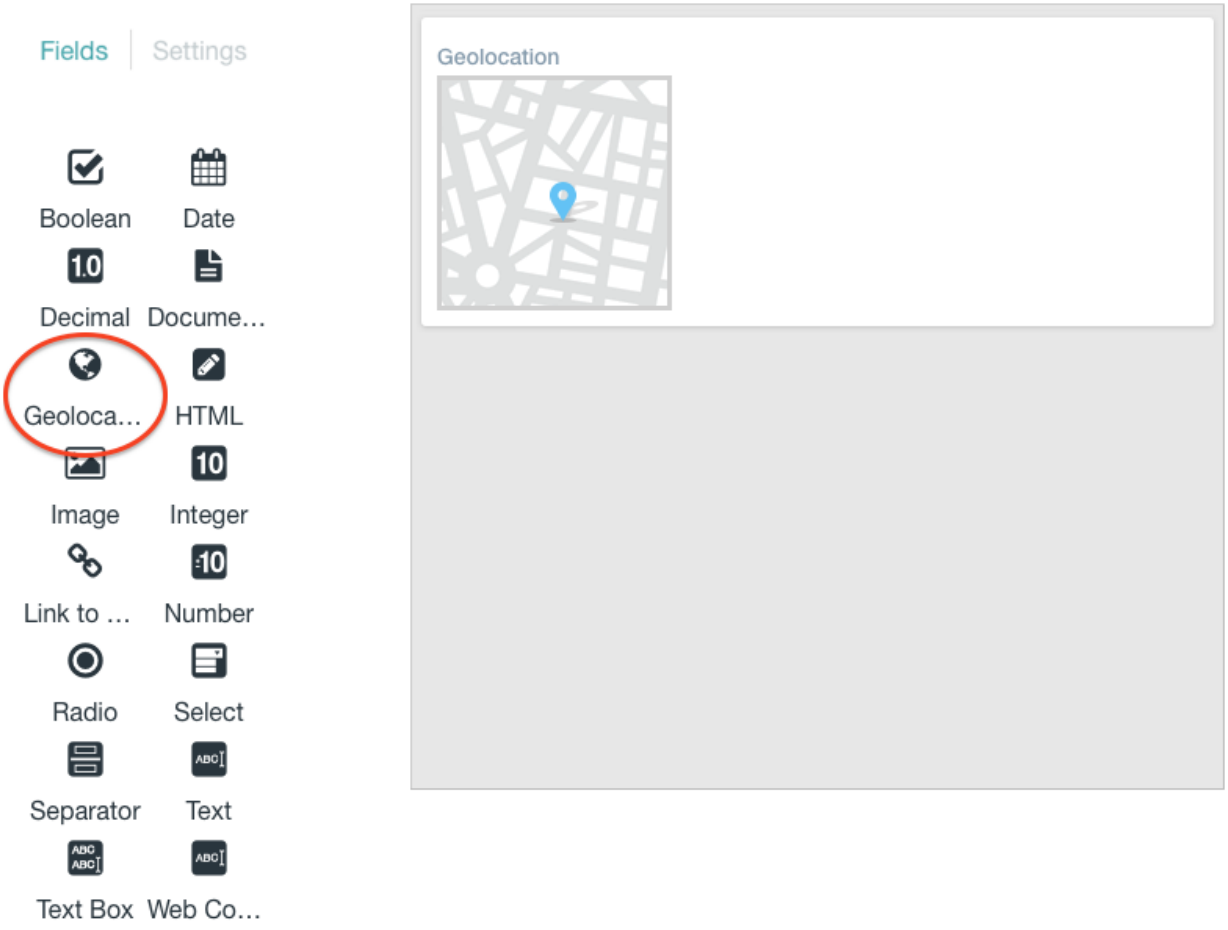


Figure 12.9: Add a geolocation field to your structure to enable geolocation in your web content.

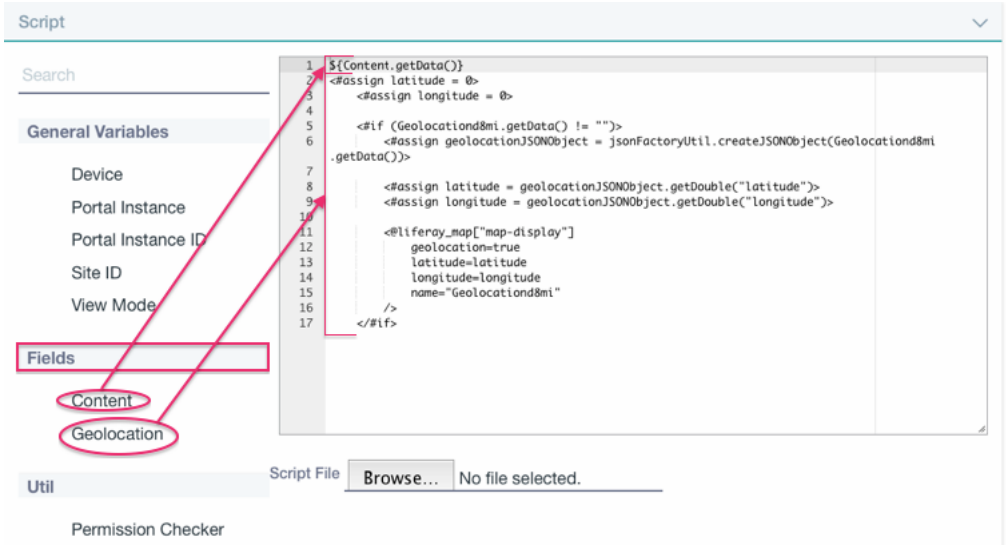


Figure 12.10: Add the Content and Geolocation snippets to quickly create your web content template.

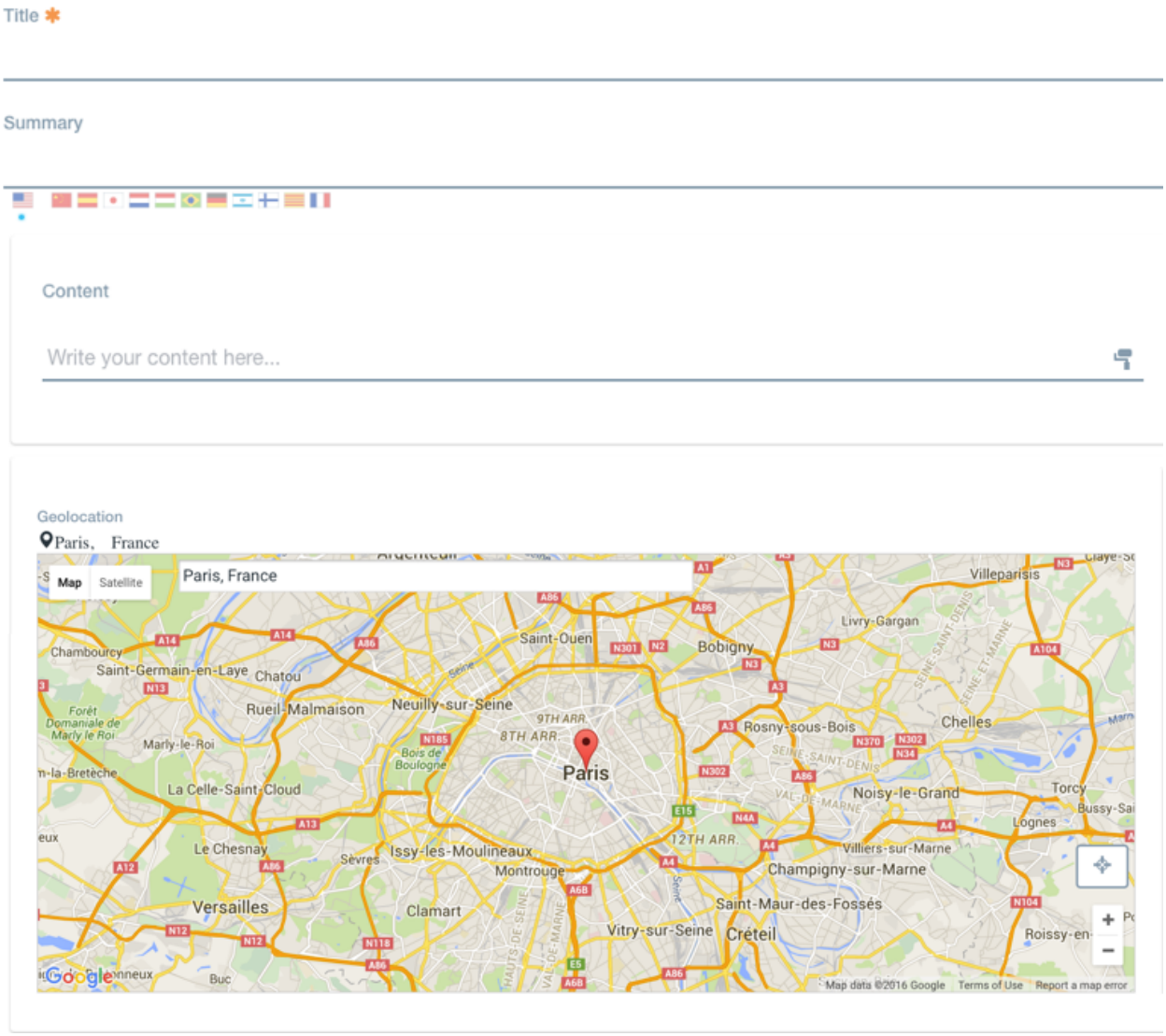


Figure 12.11: You can enter your location in the address bar, move the indicator to a location, or share your location with the browser.

your location.

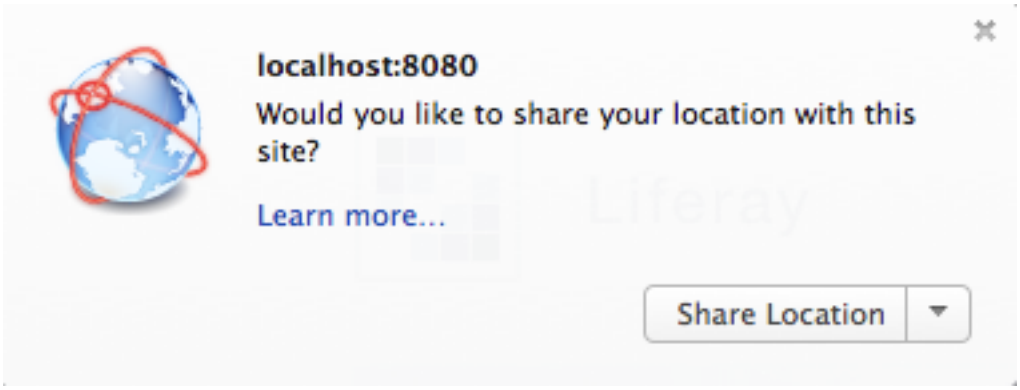


Figure 12.12: Make sure your browser is configured to share your location.

Geolocating Data Lists

To use geolocation in your dynamic data lists, you first have to create a data definition that includes a geolocation field. To do this, open the *Menu* and navigate to *Content → Dynamic Data Lists*. Click the *Options* menu and select *Manage Data Definitions*. Click the *Add* button to create a new data definition. Enter a name, optional description, and parent data definition if you have one. Scroll down and add a *Geolocation* field to the data definition, along with any other fields you wish to add and *Save*. Navigate back to the Dynamic Data Lists screen and click the *Add* button () to create a new list. Enter a name and optional description. Finally, click the *Select* button and choose the newly created data definition. Now that your data list is complete you can use the Data List Display portlet to display it.

Geolocating Documents and Media

To enable geolocation in your documents & media, you must first create a custom document type that includes geolocation metadata. You can add geolocation metadata as part of a Metadata Set or as part of the new document type. To add geolocation metadata as part of a Metadata Set, open the *Menu* () and navigate to *Content → Documents and Media*. Open the *Options* () menu, and select *Metadata Sets*. Click the *Add* () button and enter a name, optional description, and Parent Metadata Set if you have one. Scroll down and add a *Geolocation* field, along with any additional fields you wish to have, and *Save*.

To create the new document type with geolocation, navigate to *Documents and Media*, open the *Options* () menu and select *Document Types*. Click the *Add* button () and enter a name and optional description. Scroll down to the Main Metadata Fields heading and add a *Geolocation* field along with any other fields you wish to have for the document type. If you are using a Metadata Set, scroll down to the Additional Metadata Fields heading, click the *Select Metadata Set* button, and choose your Metadata Set with the geolocation metadata and *Save*.

Now, navigate back to the *Documents and Media* screen and click the *Add* button () and select your newly created document type. Fill out the information for the document, and just as with the web content, your location is automatically obtained from the browser and added to your document.

Once your assets are geolocation enabled, you can use the Asset Publisher to display the location of the assets on a map, using the map display template. Check out the Configuring Display Settings section to learn more.

Next, you'll learn about Liferay DXP's Search application.

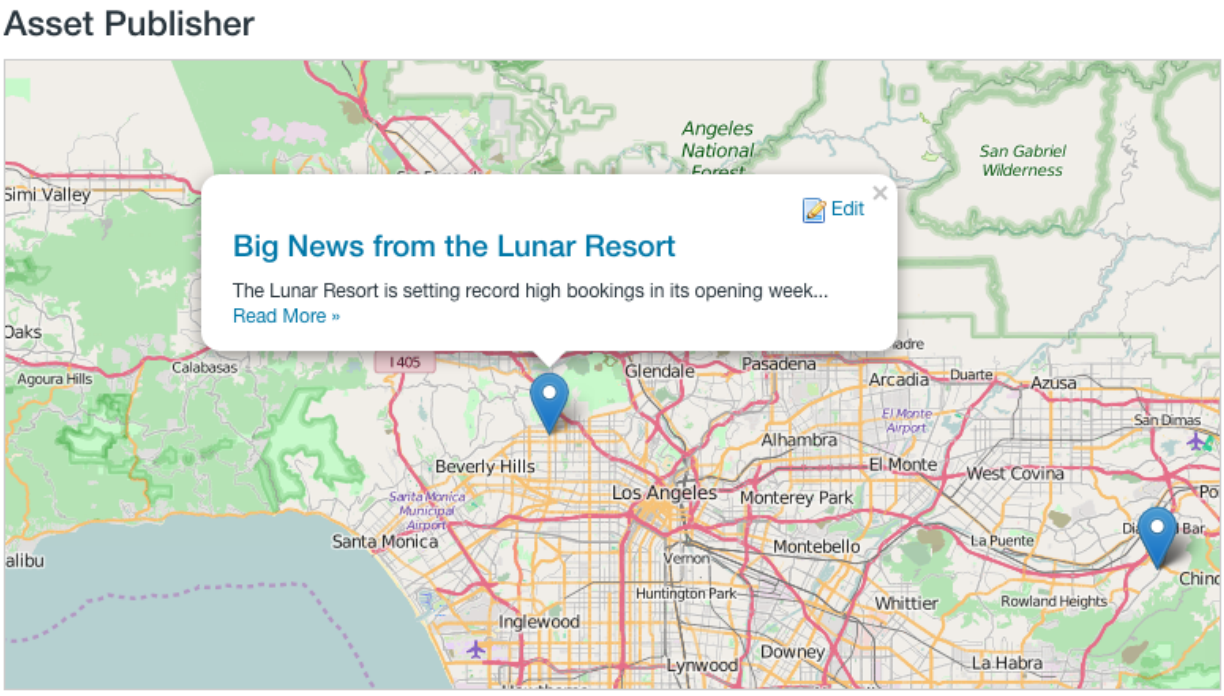


Figure 12.13: The Asset Publisher can display your geolocated assets on a map.

12.4 Publishing Assets

As you create web content, it's important to keep in mind that to Liferay DXP, the pieces of content are assets, just like message board entries and blog posts. This allows you to publish different kinds of content using Liferay DXP's Asset Publisher. You can use the Asset Publisher to publish a mixed group of various kinds of assets such as images, documents, blogs, and of course, web content. This helps in creating a more dynamic web site: you can place user-created wiki entries, blog posts, or message board messages in context with your content. You'll examine some of the Asset Publisher's features next.

Querying for Content

The Asset Publisher is a highly configurable application that lets you query for mixed types of content on the fly. By giving you the ability to control what and how content is displayed from one location, the Asset Publisher helps you to "bubble up" the most relevant content to your users.

To get to all the application's options, click the *Options* icon ( ) in the application's menu. If you click the *Configuration* option and then *Setup* (if necessary), you can configure the Asset Publisher's settings from the following three areas:

- Asset Selection
- Display Settings
- Subscriptions

Asset Selection allows you to configure which assets are displayed. You can set asset selection to either *Dynamic* or *Manual*. With dynamic asset selection, assets are automatically displayed based on certain rules

or filters. For example, you can set the Asset Publisher to display only assets of a certain type or assets to which certain tags or categories have been applied. With manual asset selection, the Asset Publisher only displays assets that have been explicitly selected by an administrator.

The Asset Publisher supports a scope that restricts both dynamic and manual asset selection. The Asset Publisher can only display assets from its configured scope. By default, the Asset Publisher app is scoped to the site of the page to which it was added. You can, however, customize the scope from the Asset Selection section of the Asset Publisher configuration window. To extend your Asset Publisher's scope, click *Select* under Scope and choose either *Global* to add the global scope, *Pages…* to add the scope to specific pages, or *Other Site…* to add the scope of anther site.

The Display Settings section of the Asset Publisher configuration window lets administrators customize many details that determine how content is displayed. The Subscription section allows administrators to enable, disable, or configure email subscriptions and RSS subscriptions. In the following sections, you'll explore the available configurations for the Asset Selection, Display Settings, and Subscriptions sections of the Asset Publisher's configuration window. You'll start by learning how select content manually. You'll see that it's very similar to using the Web Content Display application except that you can select assets of any type, not just web content articles.

Selecting Assets Manually By selecting *Manual* from the select box beneath *Asset Selection*, you're telling the Asset Publisher that you want to select content manually. You can configure multiple scopes, including the global scope, from which to select assets.

When selecting assets manually, you'll see a list of configured scopes under the Scope heading. Click the "X" button at the right to remove a scope from the list. Click the *Select* button to add additional scopes to the Asset Publisher's configuration. After you've added a scope, a new Select button appears under the Asset Entries heading. A list of assets selected for display appears in the Asset Entries section. You can select assets to be displayed by clicking on the appropriate *Select* button. One button appears for each configured scope. By default, the available asset types include the following:

- Blogs Entry
- Bookmarks Entry
- Bookmarks Folder
- Calendar Event
- Basic Document
- Google Docs
- Contract
- Marketing Banner
- Online Training
- Sales Presentation
- Documents Folder
- Dynamic Data Lists Record
- Message Boards Message
- Basic Web Content
- Web Content Folder
- Wiki Page

You can select any number of assets to be displayed. Note, however, that there's a display setting called *Number of Items to Display* that determines the maximum number of items to display (or, if pagination is enabled, the maximum number of items to display per page). The Asset Publisher enables you to mix and match different asset types in the same interface. When you're done selecting items to display, click *Save*.

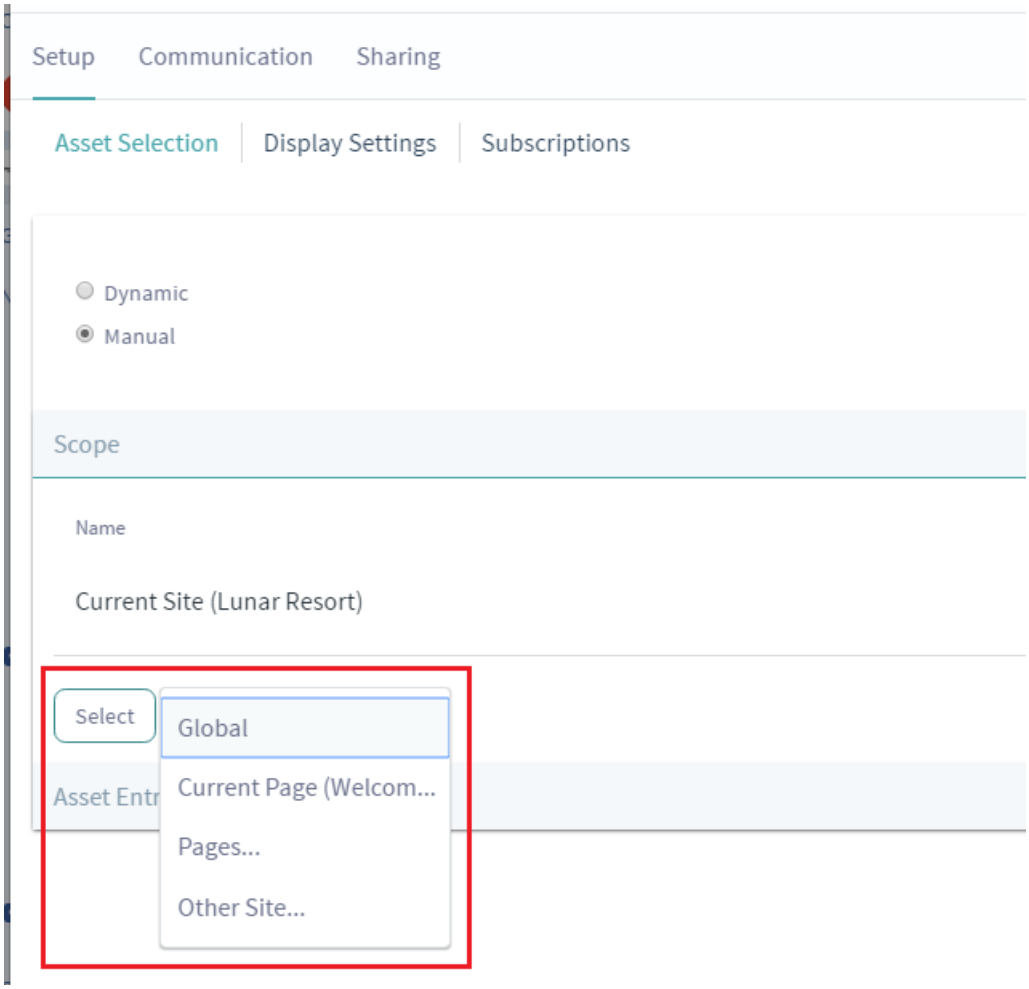


Figure 12.14: Selecting assets in the Asset Publisher manually is similar to selecting assets in the Web Content Display application except that you can select assets of any type, not just web content. You can also add scopes to expand the list of assets that are available to be displayed in the Asset Publisher.

Any selected assets are added to the list of assets that are displayed by the application. Once you have your content selected, you can configure the display types to configure how the content appears. We'll discuss the display settings in more detail after we finish discussing how to select assets for display.

While manual asset selection allows you to select assets of various types from different scopes, it can be time-consuming to periodically update the assets that should be displayed. It's often more convenient to use the Asset Publisher to select content dynamically.

Selecting Assets Dynamically The Asset Publisher's default behavior is to select assets dynamically according a set of customizable rules. These rules can be stacked on top of each other so that they compliment each other to create a nice, refined query for your content. You can define complicated rules for selecting assets for display and Liferay DXP automatically takes permissions into account. Liferay's Asset Publisher performs well in these situations since it queries by search index instead of querying the database directly. You have the following options for creating rules for selecting content:

Scope: Choose the sites from which the content should be selected. This works the same way as with manual asset selection: assets can only be displayed if they belong to a configured scope. The following scope options are available for an Asset Publisher:

- *Current Site*
- *Global*
- *Other Site*

The Other Site scope option is unavailable for Asset Publisher applications configured on a page template (e.g., Content Display Page).

Asset Type: Choose whether you'll display any assets or only assets of a specific type, such as only web content, only wiki entries, or any combination of multiple types.

Filter: Add as many filters on tags or categories as you like. You can choose whether the content must contain or must not contain any or all of the tags or categories that you enter.

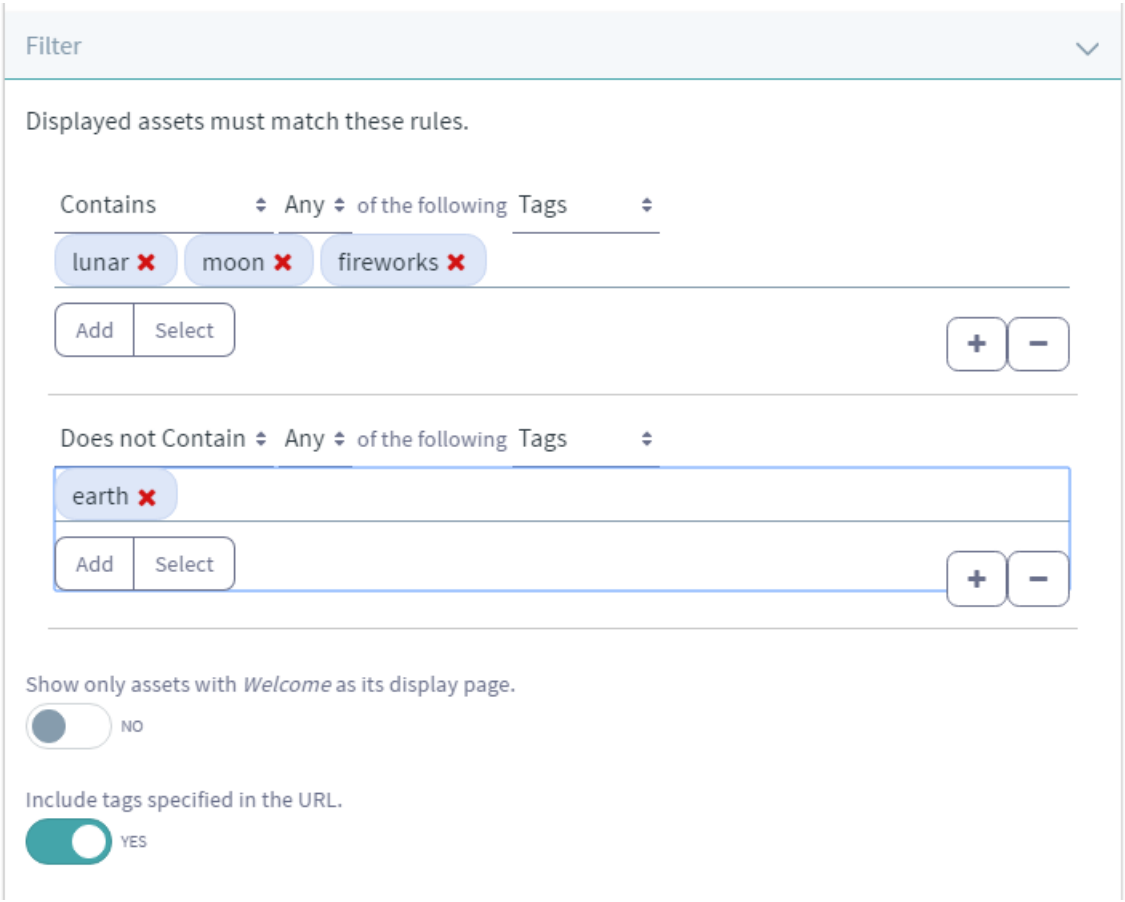


Figure 12.15: You can filter by tags and categories, and you can set up as many filter rules as you need.

Once you've set up your filter rules for dynamically selecting content, you can decide how the content will be displayed.

You can configure the Asset Publisher to display assets that only match the custom user profile attributes. This setting retrieves assets that have matching categorization. These categories must be from the global context. For example, suppose a user had a custom field called *Location* with the type *Text*. Also assume the user's location attribute is set to *Moon*. You could create a vocabulary called *Location* and a category for the Location vocabulary called *Moon*. Then you could categorize content with *Moon* in the *Location* vocabulary. With this organizational setup, adding an Asset Publisher and specifying *Location* as the Asset Publisher's custom user attribute would only display content that had been categorized as *Moon*. Pretty cool, right?

In addition, you can use these advanced filters:

- **Show only assets with *Welcome* as its display page** displays only assets specifically configured for the *Welcome* page.
- **Include tags specified in the URL?** lets you specify tags in the URL for the Asset Publisher to display.

The *Ordering and Grouping* section of the Asset Publisher lets you precisely control how content is ordered and grouped when displayed. You can order the assets displayed by Asset Publisher in ascending or descending order by the following attributes:

- Title
- Create Date
- Modified Date
- Publish Date
- Expiration Date
- Priority

For instance, suppose you have a series of "How To" articles that you want displayed in descending order based on whether the article was tagged with the *hammer* tag. Or, suppose you want a series of video captures to display in ascending order based on a category called *birds*. For these use cases, you can configure the ordering and grouping settings.

You can also configure a second ordering. The second ordering would be applied to any assets for which the first ordering wasn't sufficient. For example, suppose you chose to order assets by title and there are multiple assets with the same title. Then the second ordering would take effect. For example, you could order all the assets that had the same title by their publication dates.

You can establish grouping rules as well as ordering rules. You can group assets by type or by vocabulary. For example, suppose there's a vocabulary called *Membership Type* that belongs to your site. Suppose this vocabulary has two categories: *Premium* and *Regular*. If you group assets by Membership Type, all assets with the Premium category will be displayed in one group and all assets with the Regular category will be displayed in another group. Grouping rules are applied before any ordering rules: they're a way to divide up the displayed assets into separate lists. The ordering rules are applied separately to each group of assets.

Note that grouping and ordering rules are only one mechanism to control how your content will be displayed. You can refine the display through many other display settings which you'll examine next.

Note: The following actions will have immediate effects in your Asset Publisher: - Change the value of the *Asset Selection* option. - Change the value of the *Scope* option. - Select, add, sort or delete asset entries (only when selecting assets manually).

Other changes in the rest of the options will come into effect after clicking *Save*.

Configuring Display Settings

Open the *Display Settings* subtab of the Setup tab of the Asset Publisher's Configuration window. Here, you can configure many more settings that control the Asset Publisher's behavior and that determine how the Asset Publisher displays content. The Display Settings section gives you precise control over the display of your assets. There are many options available to configure how you want your content to appear. Many of these, such as printing, flags, ratings, comments, comment ratings, and social bookmarks work the same way they do in the Web Content Display application.

Show Add Content Button: When selected, this selector adds an *Add New* button that lets users add new assets directly from the Asset Publisher application. This is checked by default.

Display Template: This selector lets you choose an application display template to customize how the Asset Publisher displays assets. Liferay DXP creates the following display templates for each newly created site, including the default site:

- *Abstracts:* This display template shows the first 200-500 characters of the content, defined by the **Abstract Length** field. This is the default display template of the Asset Publisher.
- *Table:* This display template displays the content in an HTML table which can be styled by a theme developer.
- *Title List:* This display template displays the content's title as defined by the user who entered it.
- *Full Content:* This display template displays the entire content of the entry.

  There's also the *Rich Summary* and *Map* display templates that belong to the global scope. The Rich Summary template provides a summary view of each asset along with a *Read More* link to the article's full content. The Map template displays geolocalized assets in either a Google Map or an Open Street Map provider. The map provider can be configured in Instance Settings, and Site Settings in the Advanced section.

Abstract Length: This selector lets you select the number of characters to display for abstracts. The default is `200`.

Asset Link Behavior: The default value is *Show Full Content*. With this value selected, when the link to an asset is clicked, the full asset is displayed in the current Asset Publisher. (There's also a *View in Context* link that shows the article in the Wiki page's Wiki application.) If the value *View in a Context* is selected, clicking on an asset causes that asset to be displayed in the application to which the asset belongs. For example, a blog entry would be displayed in the Blogs application where it was created. Likewise, a forum post would be displayed in the Message Boards application where it was created. Similarly, a generic web content article would be displayed in the Asset Publisher of its configured display page. See the section below on display pages for more information.

Tip: When the Asset Publisher displays web content articles that have an associated small image, the small image becomes a link to the full article. To use this feature, add or edit a web content article that the Asset Publisher should display. Before clicking *Publish*, click on *Abstracts*, flag *Small Image*, and upload an image. Then click *Publish*. Once your web content article appears in the Asset Publisher's list, clicking the small image takes you to the full article.

Number of Items to Display: This selector lets you select the maximum number of assets that can be displayed by the Asset Publisher. If pagination, however, is enabled, there's no limit to the number of of assets that the Asset Publisher can display. So with pagination enabled, this number represents the maximum number of assets that can be displayed per page.

Pagination Type: This can be set to *None*, *Simple*, or *Regular*. With pagination set to *None*, the Asset Publisher displays at most the number of assets specified in the **Number of Items to Display** property. Setting the pagination type to *Simple* adds *Previous* and *Next* buttons that enable the user to browse through pages of assets in the Asset Publisher. Setting the pagination type to *Regular* adds more options and information including *First* and *Last* buttons, a dropdown selector for pages, the number of items per page, and the total number of results (assets being displayed).

Show Metadata Descriptions: This enables Metadata descriptions such as *Content Related to...* or *Content with tag...* to be displayed with the published assets.

Show Available Locales: Since content can be localized, you can have different versions of it based on locale. Enabling this option shows the locales available, enabling users to view the content in their language of choice.

Set as the Default Asset Publisher for This Page: The Asset Publisher app is an instanceable app; multiple Asset Publishers can be added to a page and each has an independent configuration. The default Asset Publisher for a page is the one used to display any web content associated with the page.

Enable Conversion To: If you have enabled Liferay's OpenOffice/LibreOffice integration, you can allow your users to convert the content to one of several formats:

- DOC
- ODT
- PDF
- RTF
- SXW
- TXT

  Please refer to the section on Liferay Server Administration for information on setting up Liferay's OpenOffice/LibreOffice document conversion functionality.

Enable ...: The Asset Publisher's Display Settings allow you to enable/disable the following options for displayed assets:

- Print
- Flags
- Related Assets
- Ratings
- Comments
- Comment Ratings
- Social Bookmarks

Enabling the Print option adds a *Print* link to the full view of an asset displayed in the Asset Publisher. Clicking *Print* opens a new browser window with a print view of the asset. Enabling flags, related assets, ratings, comments, comment ratings, or social bookmarks add links to the corresponding social features to the view full of the asset in the Asset Publisher.

Tip: An alternate way to add flags, comments, and ratings to a page is through the *Page Flags*, *Page Comments*, and *Page Ratings* applications. Just add the applications in the appropriate location near the asset you'd like to have feedback for.

When enabling social bookmarks, you're given sub-options to edit the display style of your social bookmarks and whether they are listed at the top or bottom of the Asset Publisher.

Show Metadata: Lets you select various metadata types to be displayed (see below). For example, you can select tags and categories for display. Upon saving your configuration, the Asset Publisher displays tags and categories for each displayed asset. Then users can click on the tags and categories to manually filter the displayed assets.

The Display Settings section of the Asset Publisher has numerous options to help you configure how your content selections are displayed to your users. Even though there are many choices, it's easy to go through the options and quickly adjust the ones that apply to your situation. You'll want to use the Asset Publisher to query for different kinds of assets in Liferay DXP that contain relevant information for your users.

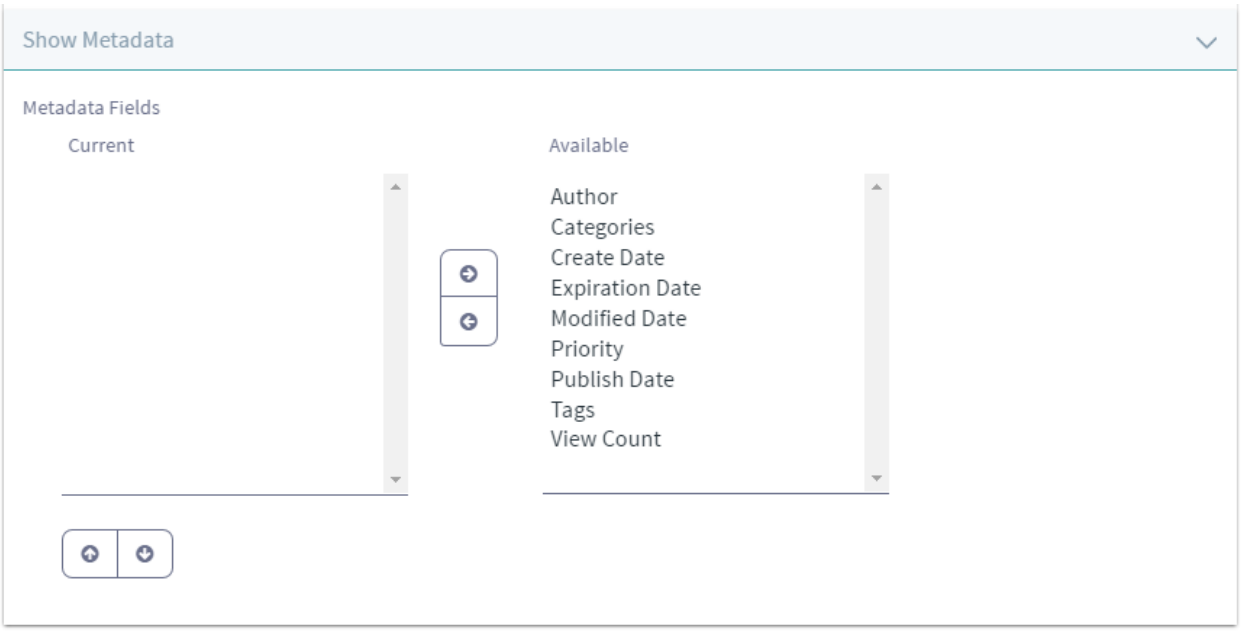


Figure 12.16: You can configure the Asset Publisher to display various kinds of metadata about the displayed assets.

Configuring Asset Publisher Subscriptions

The Asset Publisher application supports two kinds of subscriptions: RSS subscriptions and email subscriptions. To enable subscriptions, click the Asset Publisher's Options icon and select *Configuration*. In the configuration window, open the Subscriptions tab of the Setup tab. There are two options: *Enable RSS Subscription* and *Enable Email Subscription*.

Enabling RSS Subscription creates an RSS feed containing links to all of the assets that the Asset Publisher is configured to display. A link to this RSS feed appears at the bottom of the Asset Publisher application. This option is only available when the *Dynamic* Asset Selection is configured.

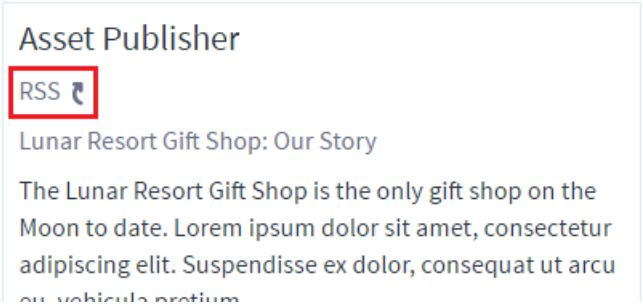


Figure 12.17: When RSS subscriptions have been enabled for an Asset Publisher application, a link to the Asset Publisher's RSS feed appears. Users can subscribe to the Asset Publisher's RSS feed using their preferred RSS reader.

Enabling Email Subscription adds a *Subscribe* link to the Asset Publisher. Users wishing to be notified of newly published assets can click on this link to be added to the subscription list. Liferay DXP periodically checks for new assets and sends emails to subscribed users informing them about the new assets. By default, Liferay performs this check every twenty-four hours.

Next, you'll look at Display Pages.

Content Display Pages

If you've been using Liferay DXP for a while, you might have noticed something about how Liferay DXP handles web content–content is never tied directly to a page. While this can be useful (because it means that you don't have to recreate content if you want to display the same thing on multiple pages), it also means that you don't have a static URL for any web content, which is bad for search engine optimization.

As an improvement, Liferay introduced the concept of *display pages* and *canonical URLs*. Each web content entry in Liferay has a canonical URL, which is the official location of the content that is referenced any time the content is displayed. A display page can be any page with an asset publisher configured to display any content associated with the page. When adding or editing web content, you can select a display page, but only pages with a configured asset publisher are available for selection.

To create a display page, you can create a page yourself, add an Asset Publisher app, and configure it yourself. Alternatively, you can use the *Content Display Page* page template included with Liferay DXP. If you're creating a Display Page manually, once you've added an Asset Publisher application to the page, open its configuration window. Then navigate to the Display Settings tab and check the *Set as the Default Asset Publisher for This Page* box. Also, for its display settings, set the Display Style to *Simple* and the Asset Link Behavior to *View in Context*.

Note: Web content linked in the Asset Publisher can be viewed by clicking their asset links. With the *View in Context* behavior checked, the link displays the web content in its configured display page. If the web content does not have a configured display page, it is displayed in the Web Content Display application to which the asset belongs.

You may now be thinking, "Wait, you just told me that each web content item has its own URL, and that this is somehow related to pages where we display a whole bunch of content on the same page?" That's right. Just watch–create a display page called *My Web Content Display Page* somewhere in Liferay using the *Content Display Page* template. Make sure the *Inherit Changes* selector is not selected. Now, on a different page, add a Web Content Display application. Click the *Add* button, enter a title and some content, click on *Display Page* at the right, and select the Display Page you just created. Then click *Publish*.

In the Asset Publisher of the *My Web Content Display Page*, click the *Read More* link to display the full content. Notice that the canonical URL for content appears in your browser's address bar. If you create your own custom display page, any additional applications that you place on the page are displayed along with the content when you access it via the canonical URL. If you used the *Content Display Page* page template for your display page, it not only features a configured Asset Publisher application but also a Tags Navigation, a Categories Navigation, and a Search application. These tools help users to quickly identify relevant content.

Next, you'll learn about publishing RSS feeds.

12.5 Publishing RSS Feeds

RSS is a family of web feed formats used to publish frequently updated works such as blog entries and news articles. RSS allows users to stay up-to-date with your site's content without actually having to visit your site! Instead, they can subscribe to your site's RSS feed with an RSS feed reader. Their RSS reader reads your site's RSS feed and displays information about all the web content that's published on your site, such as each article's title and publication date. If one of your site's web content articles grabs their attention, then they can follow their RSS reader's link to the article's full content on your site. Many RSS readers are available today, including web-based readers, ones for the Windows, Mac, and Linux platforms, and ones for mobile devices. You'll see how to create RSS feeds in Liferay DXP next.

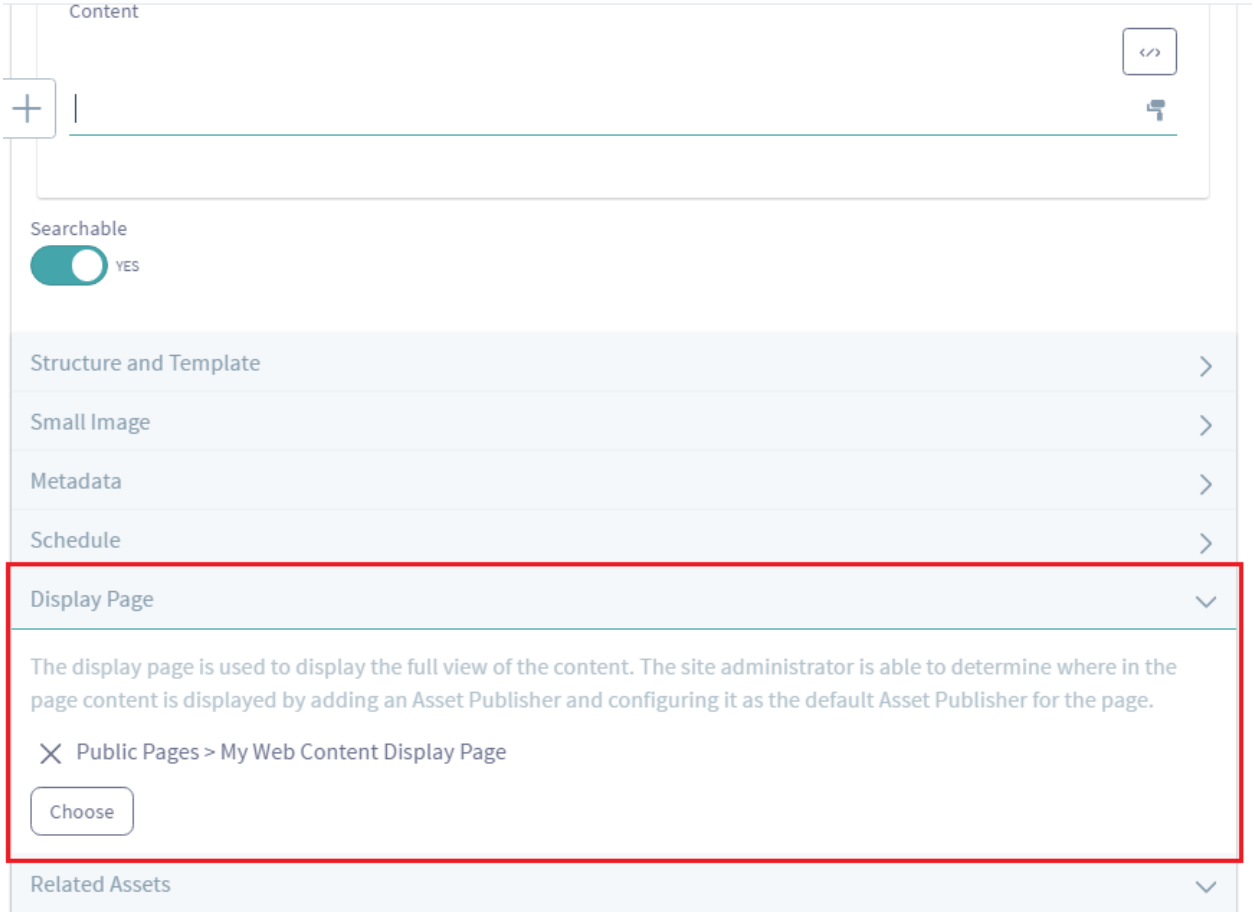


Figure 12.18: You can select a display page for a web content article when creating or editing one.

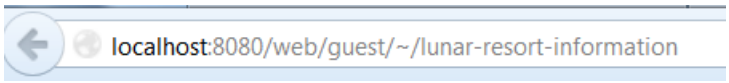


Figure 12.19: A canonical URL for a *Lunar Resort Information* page would look similar to this screenshot.

Configuring RSS Feeds

To manage a Liferay site's RSS feeds, navigate to the Site Administration → *Content* page of your site and click *Web Content*. Site administrators can use this Web Content menu option to manage their site's web content, including web content structures and templates, which you learned in the Creating Web Content section. Site administrators can also use this option to manage their site's RSS feeds. Click the *Options* icon ( ) at the top right of your screen and then *Feeds* if you'd like to add, edit, or delete RSS feeds.

Click the *Add Feed* button to add a new feed. You need to enter a name, select a target page, and select a web content structure for the feed. A feed's target page serves two purposes:

1. The site to which the target page belongs determines which web content articles appear in the feed. For example, if the target page belongs to the Marketing site, only web content articles belonging to the Marketing site will appear in the feed.

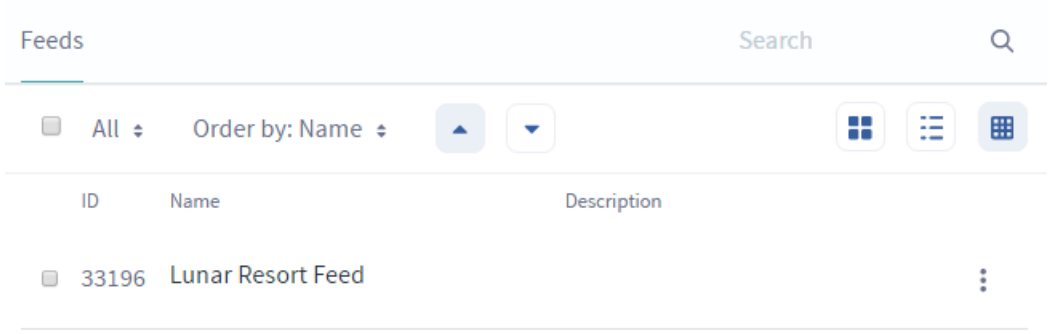


Figure 12.20: Clicking *Options → Feeds* from the Site Administration's Web Content option opens a popup window which displays your site's RSS feeds. You can add or edit RSS feeds, configure their permissions, or delete them.

2. The target page is the page where "orphaned" web content articles will be displayed. Orphaned web content articles are articles that have been published in your site but have not been configured to be displayed in specific Web Content Display applications. Liferay RSS feeds can provide links to any published web content articles, both orphaned articles and articles that have been configured to be displayed in specific Web Content Display applications. For articles that have been configured to be displayed in a specific application, the RSS feeds' links point to the Liferay page of that app. For orphaned articles, the RSS feeds' links point to the feed's target page. When users click on such links for orphaned articles, the full content of the orphaned article is displayed on the target page.

To specify a target page, you need to enter the target page's friendly URL. Note that friendly URLs do not include the host name. For example, the friendly URL of a public page called *Welcome* belonging to a site called *Marketing* might look like this: `/web/marketing/welcome`. Optionally, you can specify a target portlet ID. This would be the portlet ID of a Web Content Display application on the target page in which orphaned web content should be displayed. The application must exist or else the content will not displayed. The URL field contains the address of your RSS feed. It appears after you've actually created the feed by clicking *Save*.

The final two sections of the *Add Feed* form allow you customize which web content articles appear in your feed.

1. The Web Content Constraints section allows you to select a web content structure with which to filter the articles that appear in your feed. This is useful since all web content articles are created using web content structures.

2. The Presentation Settings section allows you to customize additional details about your feed and how articles are displayed in your feed. Leave the Feed Item Content set to *Web Content Description* if you'd just like a description of each article to appear in your feed. Set it to *Rendered Web Content: Use Default Template* if you'd like the full content of each article to appear in the feed. Customizing the Feed Type allows you to choose which web feed language to use for your feed. You can choose *Atom 1.0* (the default), *RSS 1.0*, or *RSS 2.0*. Customize the *Maximum Items to Display* to choose the maximum number of articles should appear in your feed at one time. Leave the Order By Column set to *Modified Date* to have articles arranged in order from the last time they were published or modified. You can set the Order by Column to *Display Date* if you want to have articles arranged in order from the time they were configured to be displayed in a specific Web Content Display application. Lastly, you can leave the Order by Type set to *Ascending* to have the oldest articles at the top of the feed or you can set it to *Descending* to have the newest articles at the top of the feed.

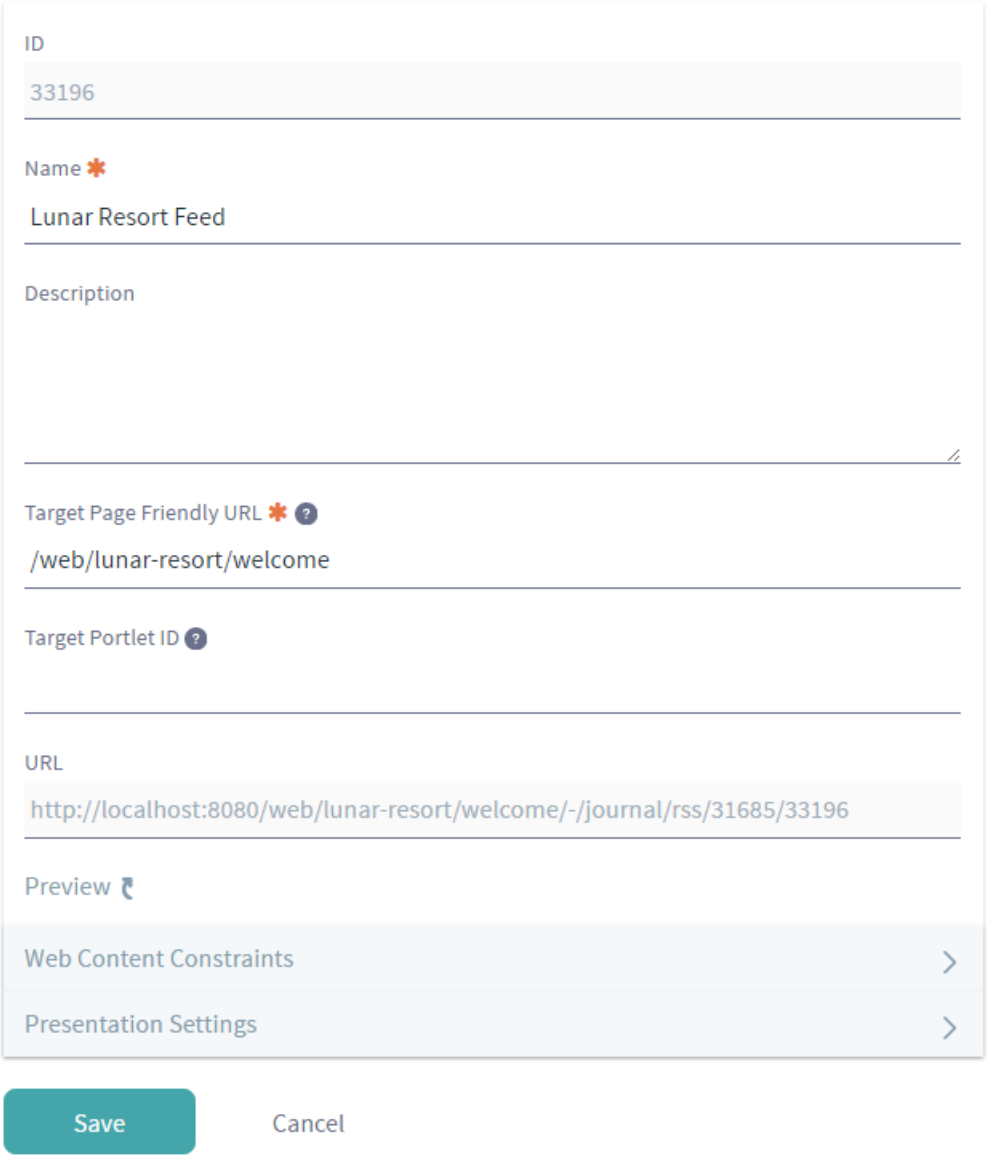


Figure 12.21: To create a new RSS feed, you only need to specify a name, target page, and web content structure. Of course, you can also configure other features of the feed such as its permissions, web content constraints, and presentation settings.

When you're done configuring your RSS feed, click *Save* to create your feed.

Once one or more feeds have been created, they'll appear in a list in the Feeds popup window when you click *Options* → *Feeds*. You can edit existing feeds using the same form used for creating them. The main difference is that when you edit an existing feed, the URL field is populated. Copy this URL into a new browser tab or window to test your feed. From the Feeds popup window, you can also customize the permissions of feeds or delete feeds.

It's possible to completely disable RSS feeds at the instance level. You can do this by setting the `rss.feeds.enabled` property to `false` in your `portal-ext.properties` file. By default, it's set to `true`. If you keep the default, RSS enabled, you can make several other RSS property customizations. Please refer to the RSS section of your `portal.properties` file for details.

Using the RSS Publisher Application

The RSS Publisher application lets you display any number of RSS feeds and configure how they are displayed. If you're looking for a web-based RSS reader, look no further: just add the RSS Publisher app to one your personal site's private pages, and voila! You have your own personal RSS reader. Open the application's Configuration menu to select the feeds to be displayed and customize the display. The RSS Publisher app can also be placed on sites' public or private pages to make feeds available to guests or site members, respectively. In these cases, make sure that only site administrators have permission to customize the RSS application and select feeds to be displayed.

Note: If you are running your server behind a proxy, you must set the appropriate Java proxy settings (such as `http.proxyHost=` and `http.proxyPort=`) in your `setenv` script, or in your `system-ext.properties`. Without these properties, the RSS Publisher application can't access any RSS feeds.

Once you've added the RSS Publisher app to a page, open the application's Configuration menu by clicking on the *Options* icon () at the top right corner of the application and selecting *Configuration*.

By default, the RSS application displays one feed. In the Feeds section, click on the plus sign to add a new feed or on the minus sign to remove a feed. Enter the URL of the RSS feed to display into the URL field. If you leave the Title field blank, the feed's title appears in the RSS application. If you enter a custom title into the Title field, the custom title appears instead of the feed's title.

In the top section, use the following selector buttons to select the feed details that should be displayed:

- Show Feed Title
- Show Feed Published Date
- Show Feed Description
- Show Feed Image
- Show Feed Item Author

You can also select the number of entries and expanded entries that should be displayed per feed. Expanded entries show more of an article's actual content than regular entries. By default, each feed shows four entries per feed and eight expanded entries per feed. You can set the feed image alignment to control whether feed images appear to the right or left of the text. By default, the feed image alignment is set to *Right*.

You've learned how to create, manage, and use RSS feeds. Great job!

In this chapter, you explored Liferay DXP's asset framework. Any type of content in Liferay is considered an asset and can utilize the features provided by the asset framework: tags, categories, comments, ratings, and relationships. You examined the Asset Publisher application and looked at the many configuration

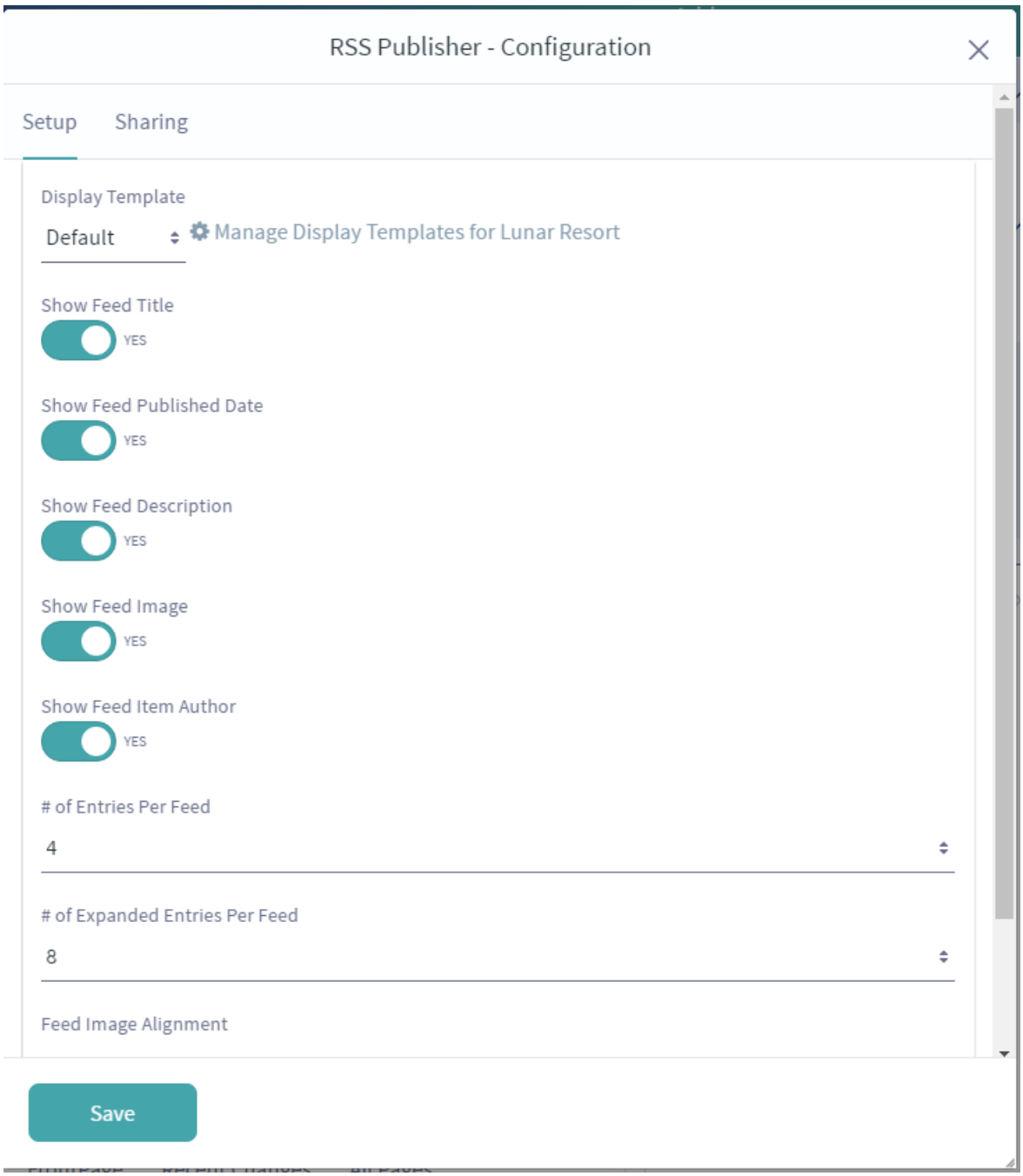


Figure 12.22: The RSS Publisher app's configuration window lets you choose feeds to be displayed and allows you to customize the display settings.

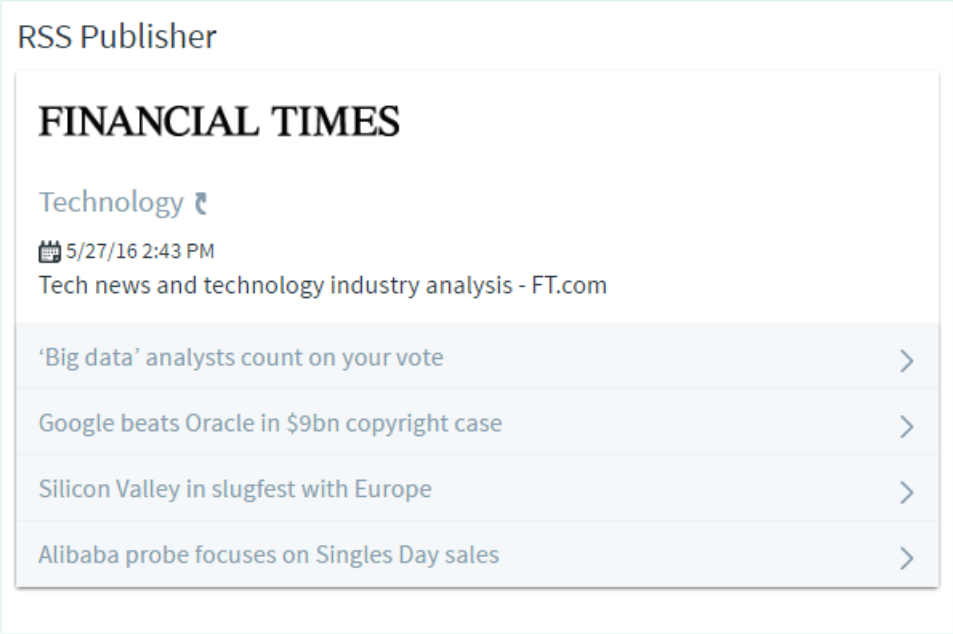


Figure 12.23: By default, the RSS Publisher app is configured to display feeds from the Financial Time. This image displays what the Financial Times feed looks like in the RSS Publisher app.

options for choosing what kinds of assets to display and how to display them. You saw that the Asset Publisher application is designed to integrate with the Tags Navigation and Categories navigation applications to allow users to browse content more easily. You also learned about the Display Page attribute of web content, the Content Display Page page template, and canonical URLs for assets. Assets can have display page associated with them so that the full view of the asset is displayed on the display page. The display page of an asset is used in the asset's canonical URL.

12.6 Restoring Deleted Assets

Have you ever had that life-altering experience where you deleted an important file and immediately regretted deleting it? The deed is usually followed by a palm to the forehead or a sick feeling. Good news! Liferay DXP is here to turn that frown upside down with the *Recycle Bin* feature. With the Recycle Bin, the *Move to the Recycle Bin* action replaces *Delete* for certain asset types. Content is now temporarily stored in the Recycle Bin. This allows the content to be restored back to its original state. Recycled items can expire after a certain period of time, resulting in their permanent deletion. Before diving into how the Recycle Bin works, you'll look at how to configure it.

Configuring the Recycle Bin

The Recycle Bin supports instance-wide scope or site-specific scope. The instance-wide scope of the Recycle Bin is set by adding the `trash.enabled` property to your `portal-ext.properties` file. By default, the Recycle Bin is enabled instance-wide. You'll go into more detail for adding this property and several others to your properties file later in the section. First, you'll explore the UI and see what the Recycle Bin can do.

First, you'll configure the Recycle Bin for site-specific scoping. Choose the site you'd like configure for the Recycle Bin from the Site Administration menu. Then click *Configuration* → *Site Settings*. Next, select the top *Advanced* tab and click *Recycle Bin*. You'll notice a few configurable options:

Enable Recycle Bin: enable and disable settings for the Recycle Bin's site-specific scope.

Trash Entries Max Age: customize the number of minutes a file is kept in the Recycle Bin until its permanent deletion (default is 43200 minutes, or 30 days).

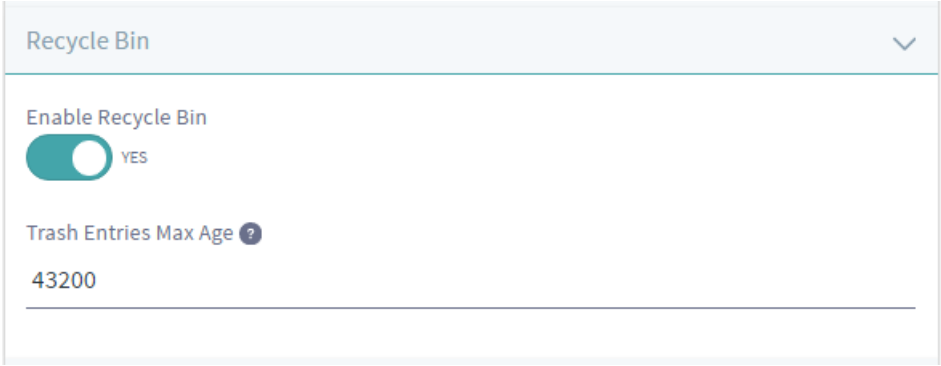


Figure 12.24: The Recycle Bin offers several configurable options for your site.

When you've finished configuring your Recycle Bin settings, click *Save*.

Note: If you disable the Recycle Bin while it's still holding recycled items, the recycled items reappear in the Recycle Bin if it is re-enabled.

You can also configure the Recycle Bin via properties in the `portal.properties` file. Remember that it's a best practice not to edit the `portal.properties` directly, but to create a separate `portal-ext.properties` file containing the properties to override. There are some additional options not available in the GUI that you can set:

`trash.search.limit=500`: set the limit for results used when performing searches in the Recycle Bin (default is 500).

`trash.entry.check.interval=60`: set the interval in minutes for how often the trash handler runs to delete trash entries that have been in the Recycle Bin longer than the maximum age (default is 60).

Also, as was mentioned earlier, there are properties to enable the Recycle bin instance-wide and set trash entries' maximum age.

`trash.enabled=true`: set this property to *false* to disable the Recycle Bin for all sites in the portal (default is *true*).

`trash.entries.max.age=43200`: set the number of minutes trash entries should be held before being permanently deleted.

Visit the portal.properties file to view all of the configurable properties for the Recycle Bin.

Next, you should make sure permissions are set properly for users who can handle/view the assets in the Recycle Bin. Users who had *View* permissions on a document when it was recycled can also view that document in the Recycle Bin. Users who had *Update* or *Delete* permissions on a document when it was recycled can restore the document.

Now that you've successfully configured the Recycle Bin, you'll look at how to use it.

Using the Recycle Bin

The Recycle Bin is temporary storage configured for multiple asset types across you Liferay instance. Instead of offering a specific Recycle Bin for each asset type, Liferay DXP provides a central master Recycle Bin where different asset types can be stored. This provides an easy search and recovery process.

You can recycle several different types of assets, including:

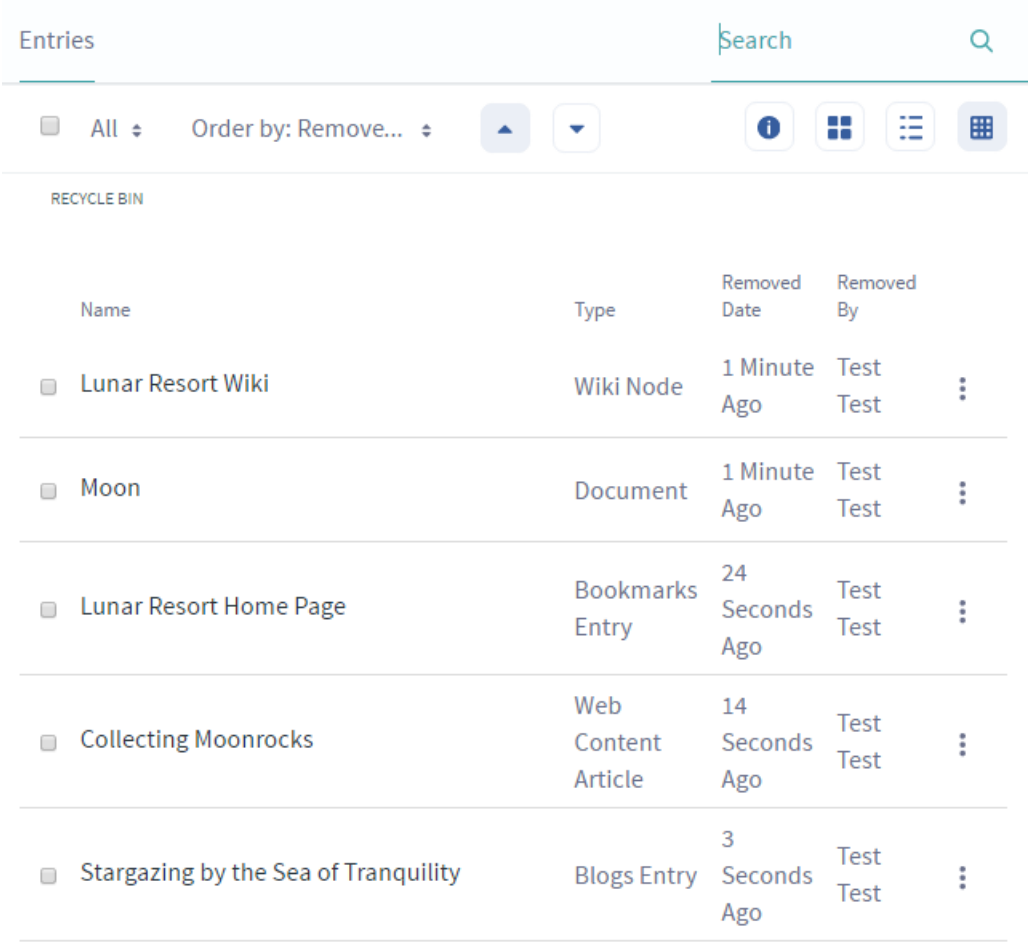


Figure 12.25: The Recycle Bin provides a seamless administrative experience for deleting and removing content.

- Blogs
- Bookmarks
- Documents and Media
- Message Boards (and attachments)
- Web Content
- Wiki (and attachments)

Note: The wiki and Message Board attachments are stored in a separate Recycle Bin instance unique to the attachment's app. For instance, when moving a wiki attachment to the Recycle Bin, it can only be restored from the Wiki app's *Removed Attachments* Menu, not the site-scoped Recycle Bin.

For a quick example to show how easy the Recycle Bin is to use, you'll practice sending a web content article to the Recycle Bin and then restoring it. You'll run through two different methods of restoring the file.

1. Navigate to Site Administration and select *Content → Web Content*.
2. Select the *Add* button ( ) and click *Basic Web Content*.
3. Enter some text for the Title and Content and click *Publish*.

4. Click the article's *Actions* button (⋮) and click *Move to the Recycle Bin*.

   Note that the *Delete* button is not listed. Liferay DXP avoids the risk of accidental deletion of your files by funneling the content through the Recycle Bin.

5. After deleting the file, a success message appears, offering an *Undo* option. Click *Undo*. The web content is retrieved from the Recycle Bin and stored in its original place.

6. Select the *Move to the Recycle Bin* button again.

7. Navigate back to Site Administration and click the Recycle Bin button from the Content dropdown.

8. Find your sample web content and click its *Actions* button.

9. You can restore or delete the content. Select *Restore*.

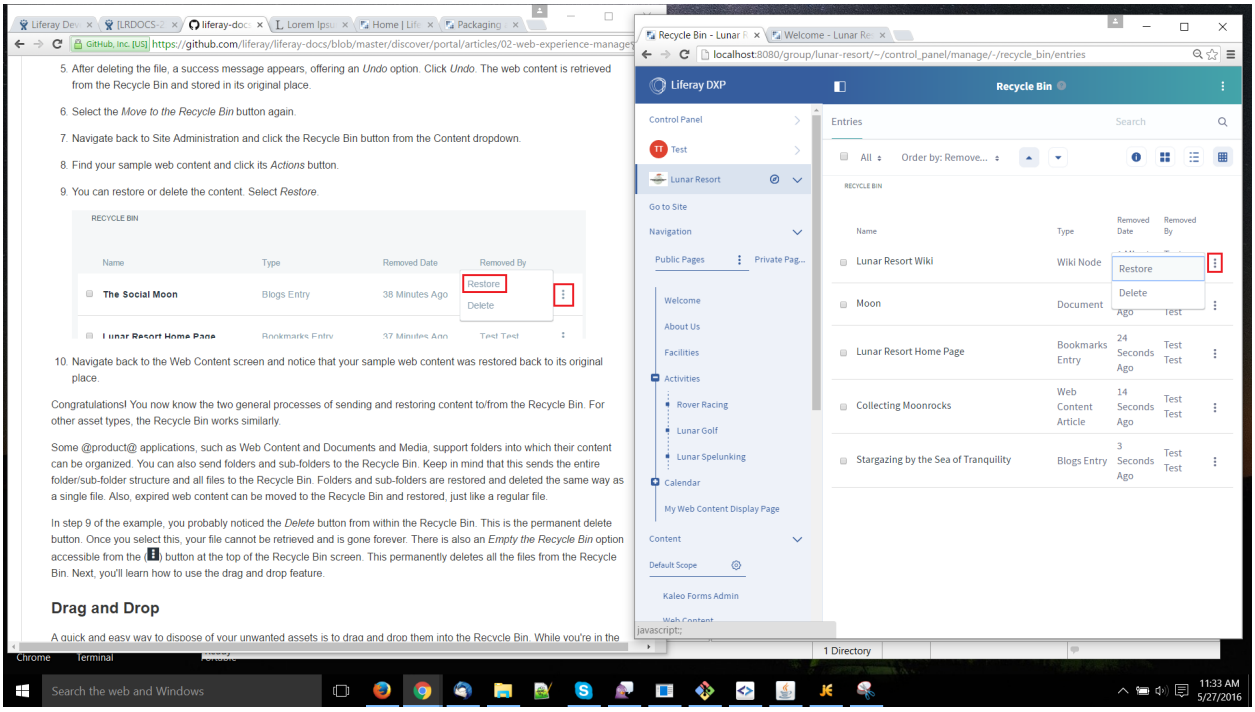


Figure 12.26: In the Recycle Bin, you have the option of restoring or permanently deleting the content.

10. Navigate back to the Web Content screen and notice that your sample web content was restored back to its original place.

Congratulations! You now know the two general processes of sending and restoring content to/from the Recycle Bin. For other asset types, the Recycle Bin works similarly.

Some Liferay DXP applications, such as Web Content and Documents and Media, support folders into which their content can be organized. You can also send folders and sub-folders to the Recycle Bin. Keep in mind that this sends the entire folder/sub-folder structure and all files to the Recycle Bin. Folders and sub-folders are restored and deleted the same way as a single file. Also, expired web content can be moved to the Recycle Bin and restored, just like a regular file.

In step 9 of the example, you probably noticed the *Delete* button from within the Recycle Bin. This is the permanent delete button. Once you select this, your file cannot be retrieved and is gone forever. There is also an *Empty the Recycle Bin* option accessible from the ( ) button at the top of the Recycle Bin screen. This permanently deletes all the files from the Recycle Bin. Next, you'll learn how to use the drag and drop feature.

Drag and Drop A quick and easy way to dispose of your unwanted assets is to drag and drop them into the Recycle Bin. While you're in the Control Panel, you can simply select an asset and drag it to the Recycle Bin portlet located on the Control Panel menu. When you click and begin dragging the asset, a message appears near your cursor notifying you of the amount of files ready to be moved, and the Recycle Bin is highlighted, showing you where the files can be dropped. After you drop the asset onto the Recycle Bin portlet, the asset is removed from its original location and transferred to the Recycle Bin.

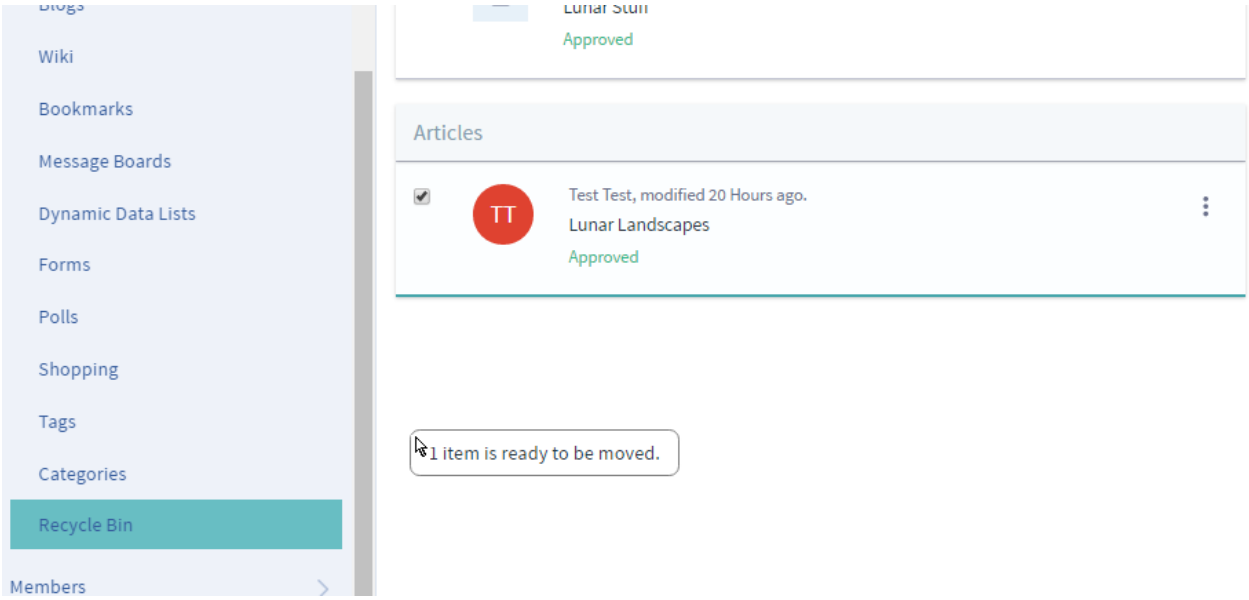


Figure 12.27: A quick and easy way of disposing your items is the drag and drop method.

Next, you'll explore the Recycle Bin's intelligence and behind the scenes support that aids in a seamless recycling experience.

Recycle Bin Intelligence and Support Have you ever wondered if it's possible to check the IQ of a software feature? Unfortunately, there is no tangible way to do this; however, if there were, the Liferay Recycle Bin would be at the top of its class. As you've learned already, it supports multiple asset types, a drag and drop feature, an Undo option, and many more. Have you ever wondered what happens to file shortcuts if their linked assets are recycled? What if you restore a file that has the same name as another file currently stored in your site/instance? The Recycle Bin already knows how to handle these types of issues for a seamless user experience.

When documents with shortcuts are moved to the Recycle Bin, the shortcuts are invalidated. This ensures Liferay DXP has active links and shortcuts at all times. This cuts down on maintenance time and backtracking after recycling an important asset.

Another important trait is the duplicate name recognition feature. When a file is restored, the Recycle Bin scans the corresponding asset type files currently in the site/instance to check for duplicate file names.

If a duplicate file name is found, the Recycle Bin prompts you to overwrite the existing file or rename the file name you're trying to restore.

For example, suppose you have the document `file1` stored in the Recycle Bin and you have a separate document you created later with the same name in the document library. If you try to restore the `file1` document, the Recycle Bin recognizes duplicate names and prompts you to overwrite the existing document in the document library or rename the document you're trying to restore.

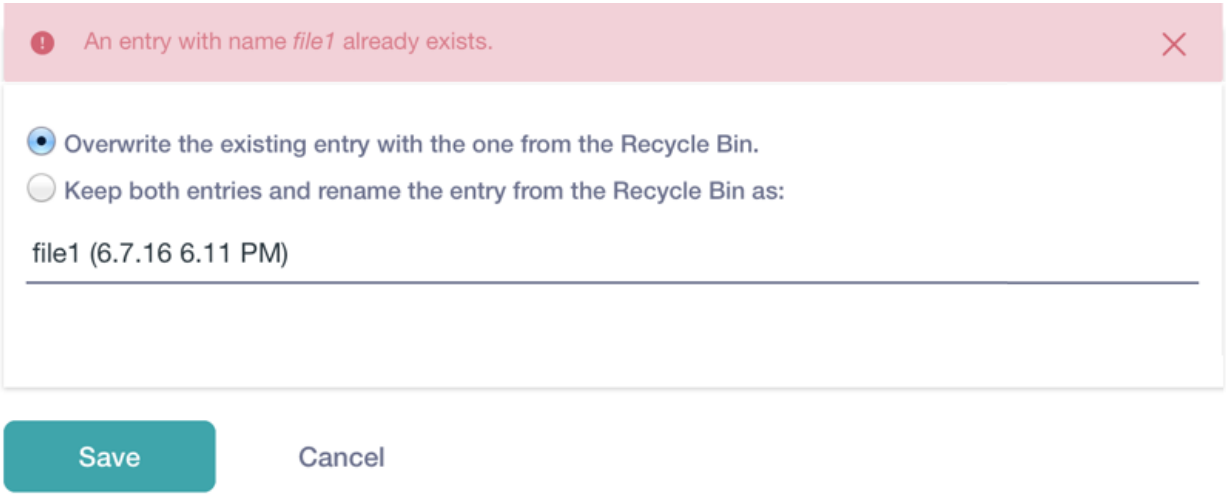


Figure 12.28: The Recycle Bin always scans your site/instance for duplicate file names during the restoration process.

Although the Recycle Bin prohibits the restoration of files that match pre-existing file names in your site/instance, it will store files with matching names.

Have you thought about how the Recycle Bin works during the staging process? Although you learned earlier that there is only one master Recycle Bin, the staging process requires a bit more flexibility with the Recycle Bin to maximize its productivity. Therefore, when staging is enabled, there is a new and separate Recycle Bin: the *Staging* Recycle Bin. The original Recycle Bin, or *Live* Recycle Bin, holding unstaged material is still viewable while in staging; however, it is never used.

During staging, everything you recycle is sent to the Staging Recycle Bin. This prevents staged and unstaged recycled content from mixing. Do you see why this would be a problem? Consider you have an unstaged document currently on your live site. Next, you enable staging and delete that document. If you were to turn staging off and return to the live site, without separate Recycle Bins, the live document would be located on your site and in the Recycle Bin! Because of this, the separate Staging Recycle Bin is necessary and only used during the staging process. Finally, when you publish your staged material to live, the Staging Recycle Bin content is transferred to the Live Recycle Bin.

Note: The Staging Recycle Bin saves its contents until the staged material has been published to the live site. This means that you can turn the staging mode on and off without losing your recycled material.

The Recycle Bin saves you time by letting you restore content that's been recycled. Instead of recreating or re-uploading content, you'll be tailoring your Liferay instance to fully leverage its capabilities.

Chapter 13

Configuring Applications

Just like siblings have common features inherited from their parents, applications that ship with Liferay DXP also share common features. These include look and feel, exporting/importing app data, communication, sharing, permissions, scoping, and configuration templates. You'll learn how these features work together to facilitate information flow within Liferay DXP and provide an enhanced experience for your users. So before you get into the nitty gritty of the applications themselves, it's best to cover these common features first, starting with the look and feel configuration options.

13.1 Look and Feel Configuration

An administrator can access the look and feel configuration menu of any Liferay DXP application by clicking on the *Options* icon ( ) at the top right corner of the app and selecting *Look and Feel Configuration*. The location of the Options icon may vary, depending on your theme. Liferay apps' look and feel dialog boxes contain six tabs:

- General
- Text Styles
- Background Styles
- Border Styles
- Margin and Padding
- Advanced Styling

After making customizations on any tab, remember to click the *Save* button to apply your changes. To see the effect of your changes, you may have to refresh the page. If you don't like the effect of your changes, click the *Reset* button to discard them.

On the General tab, you can enable the *Use Custom Title* selector to rename your app's title. The value you enter in the title box will be displayed at the top of the app's window on the page. You can also select a language from the app title drop-down menu. If you've provided a language key translation for the language you select, then your app's title will be displayed in the selected language.

If you select a page in the *Link Portlet URLs to Page* drop-down menu, all URLs inside the portlet link to the context of the page you selected. This includes pagination links created by the portlet. The current page is the default. This is intended to be an elegant way to show web content articles' full view from the Asset Publisher. If you use the View in a Specific Portlet feature or web content articles' Display Page attribute, you

Figure 13.1: To illustrate using a custom title, the Asset Publisher's default title has been replaced with a more appropriate heading.

must be careful since it redirects all links and not just links to assets. See the Configuring Display Settings article for more details.

You can also choose whether or not to apply application decorators to your app. Decorators can be used to add contrast between your app and the page. There are three decorators available: *Barebone*, *Borderless*, and *Decorate*. The Decorate application decorator is applied by default. Be careful about turning app borders off; some themes assume that app borders are turned on and may not display correctly with them turned off.

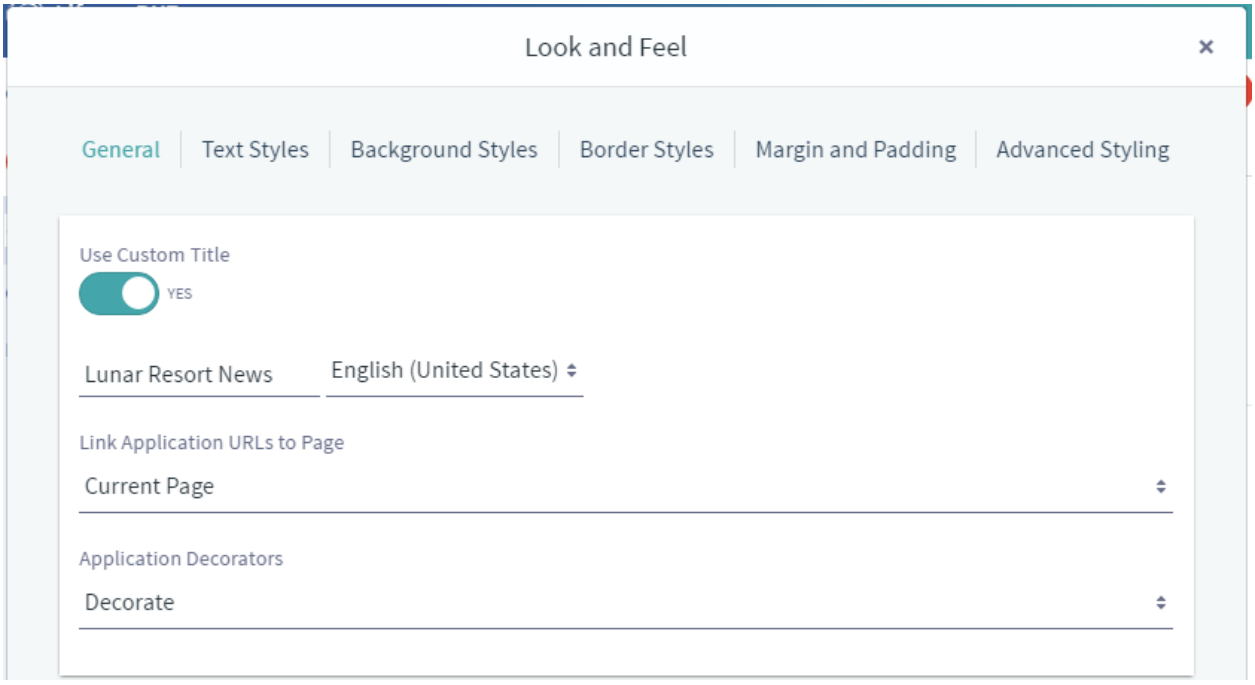


Figure 13.2: The General tab of the Look and Feel Configuration menu allows you to define a custom app title, link app URLs to a specific page, and select the app contrast option using decorators.

The Text Styles tab allows you to configure the format of the text that appears in the app. The fonts you can choose from include Arial, Georgia, Times New Roman, Tahoma, Trebuchet MS, and Verdana. Arial is the default. You can set the text to bold, italics, or both. You can set the font size anywhere from 0.1 em to 12 em, with 0.1 em increments. 1 em is the default. You can set the text color to any six digit hex color code. If you'd like help choosing a color, click on the text box to open the color palette. You can set the text alignment

to Left, Center, Right, or Justified. (Justified text is both left and right aligned.) You can set an Underline, Overline, or Strikethrough as the text decoration. The default text decoration is None.

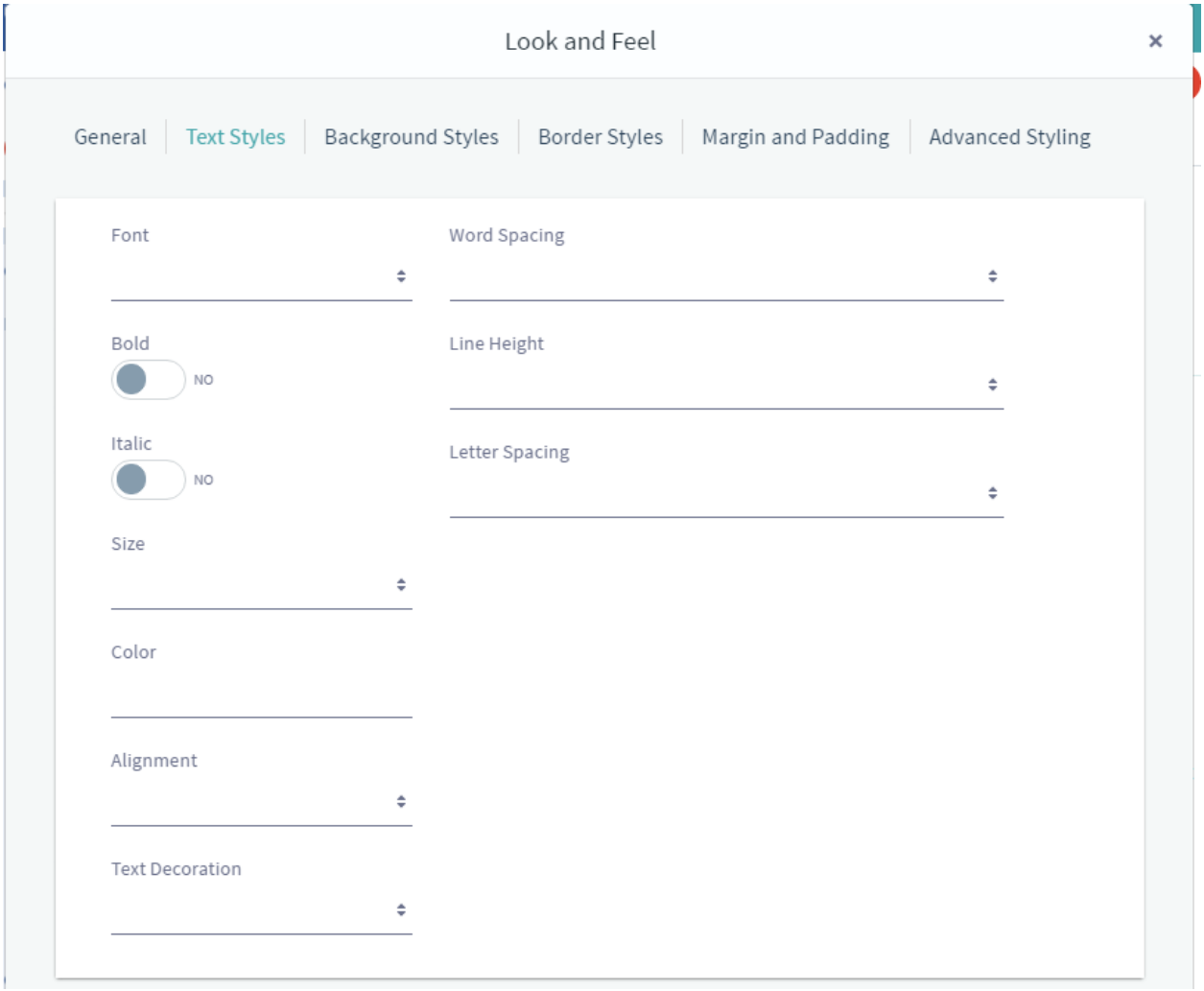


Figure 13.3: The Text Styles tab lets you configure the format of the text that appears in the app.

You can set the word spacing anywhere from -1 em to 0.95 em, with 0.05 em increments. 0 em is the default. You can set the line height anywhere from 0 em to 12 em, with 0.1 em increments. 0 em is the default. Finally, you can set the letter spacing anywhere from -10 px to 50 px, with 1 px increments. 0 px is the default.

The Background Styles tab allows you to specify the app's background color. When you select the text space, you're given a color palette to choose your background color or you can manually enter any six digit hex color code.

On the Border Styles tab, you can configure your app's border width, style, and color. For each of these attributes, leave the *Same for All* selector enabled to apply the same settings to top, right, bottom, and left borders.

For border width, you can specify any % value, em value, or px value. For border style, you can select Dashed, Double, Dotted, Groove, Hidden, Inset, Outset, Ridge, or Solid. For border color, you can enter any six digit hex color code, just like for the text color and background color. You can also use the color palette.

The Margin and Padding tab allows you to specify margin and padding lengths for the edges of your app.

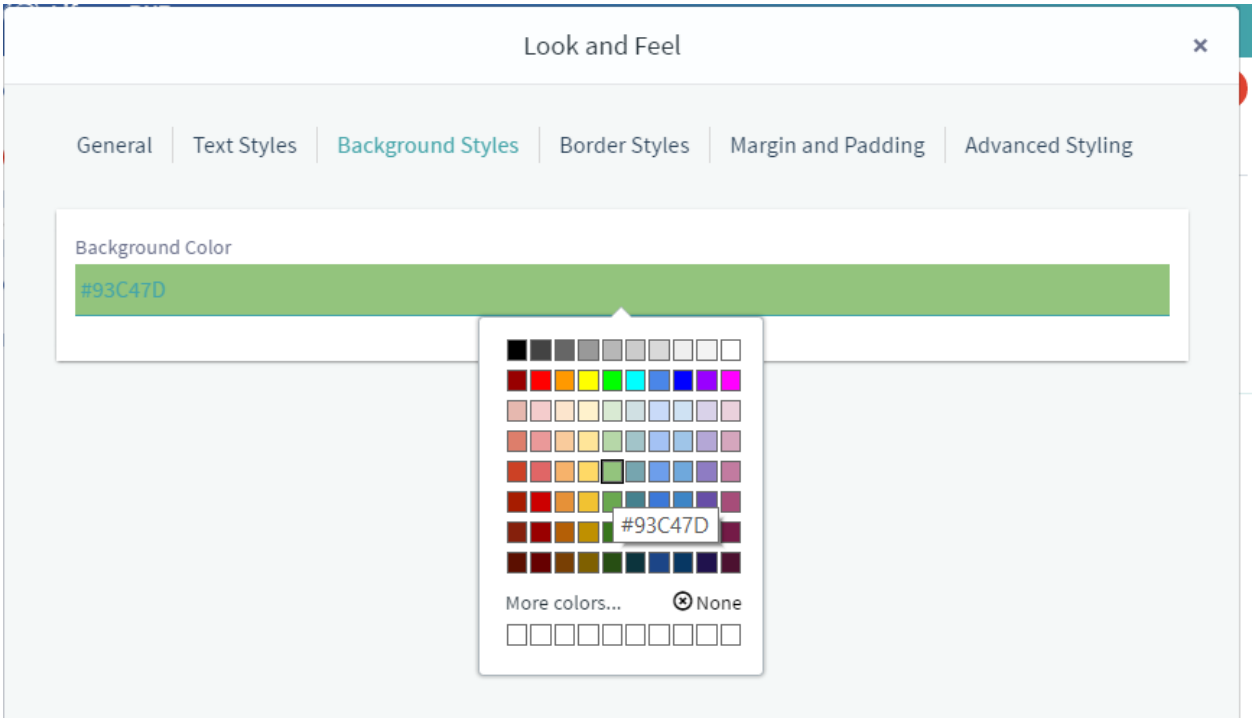


Figure 13.4: The Background Styles tab lets you specify the app's background color.

Just like for border styles, leave the *Same for All* selector enabled to apply the same settings to each side (top, right, bottom, and left) of the app.

For both padding and margin, you can specify any % value, em value, or px value.

The Advanced Styling tab displays current information about your app, including your app's Liferay ID and CSS classes.

On this tab, you can also enter custom CSS class names for your app and custom CSS code. Clicking the *Add a CSS rule for just this portlet* or *Add a CSS rule for all portlets like this one* links adds the CSS code shells into your custom CSS text box. If you check the *Update my styles as I type* box, your CSS code will be dynamically applied to your app so you can see the effects of your edits.

Next, you'll learn about exporting and importing app data.

13.2 Exporting/Importing App Data

There are times when you want to export/import specific content created from an application, and not deal with site data at all. Many Liferay apps allow you to export or import their app data indiviually. These include many of Liferay's collaborative applications, such as the Blogs, Wiki, and Message Boards apps. To export or import app data, click on the *Options* icon ( ) of your app and select *Export/Import*. Exporting app data produces a `.lar` file that you can save and import into another application of the same type. To import app data, you must select a `.lar` file. Be careful not to confuse app-specific `.lar` files with site-specific `.lar` files. See the Importing/Exporting Pages and Content section for a discussion of importing and exporting site page data.

You can export/import app content two ways. You can navigate to the app's administrative area located in the Menu, or you can visit the indiviual app that resides on a Liferay page. Both export/import menus work

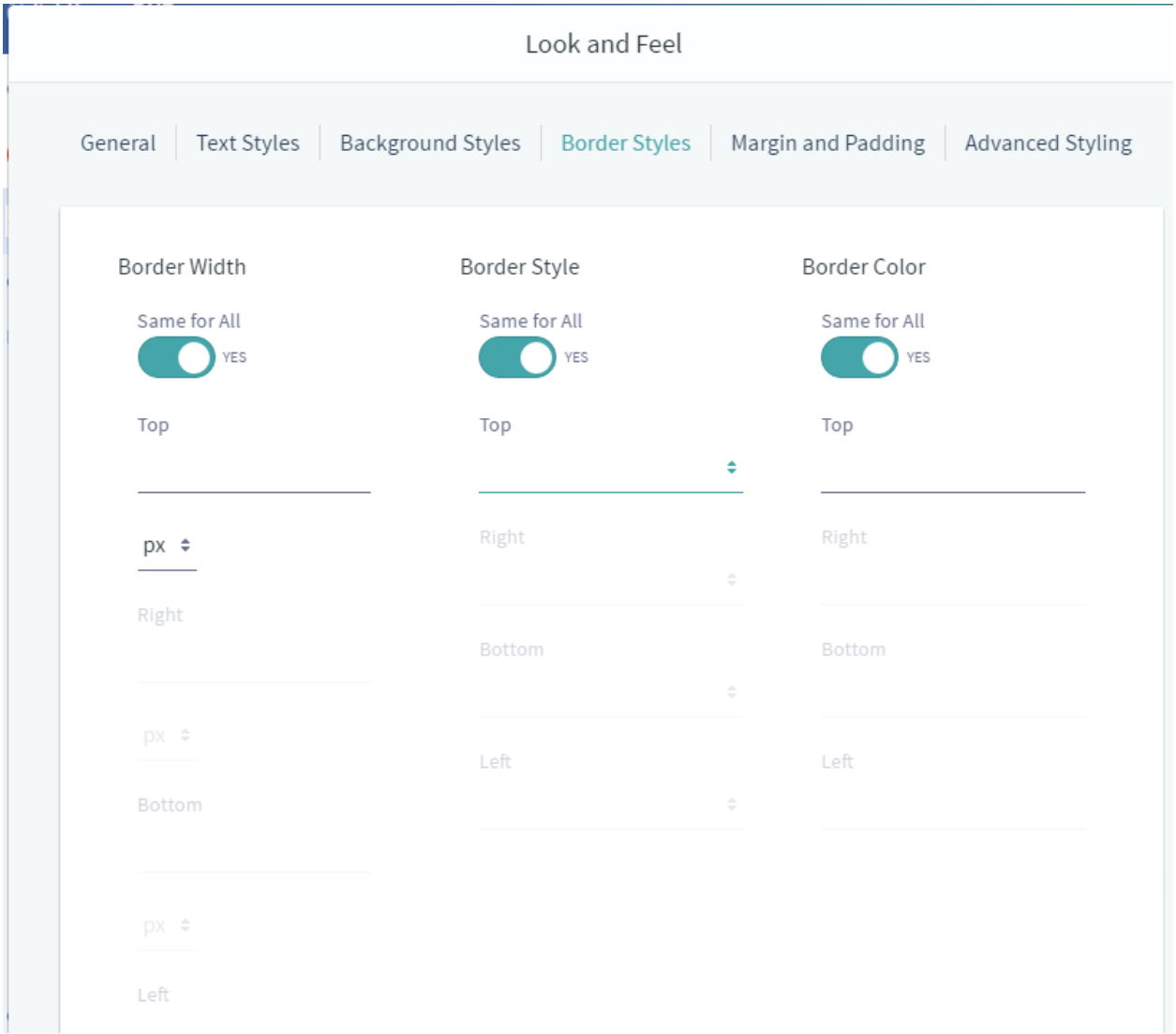


Figure 13.5: The Border Styles tab lets you specify a border width, style, and color for each side of the app.

the same, but the administrative area may hold different content than its indiviual app counterpart (e.g., Web Content Admin in Menu and Web Content Display app will not offer same content for export/import), so be wary of your app selection. To reach the Export/Import menu for an app listed in the Menu, navigate to the app in the Menu, and then select the *Options* icon ( ) from the top right of the page and click *Export/Import*. Likewise, you can navigate to the Export/Import menu for an individual app by navigating to the app's Options menu and selecting *Export/Import*.

You'll explore the export process for apps first.

First, you can choose to export your application's configuration settings. This exports your customized settings that you've configured from your application's *Options* → *Configuration* menu.

Next, you can select a *Date Range* of content that you'd like to export. Content that has been added to your app within your specified date range is included in the `.lar` file. You also have options of choosing *All* dates or just the most recently added content, or *Last*.

By checking the *Content* box, you can choose specific content you'd like to export. When you check the

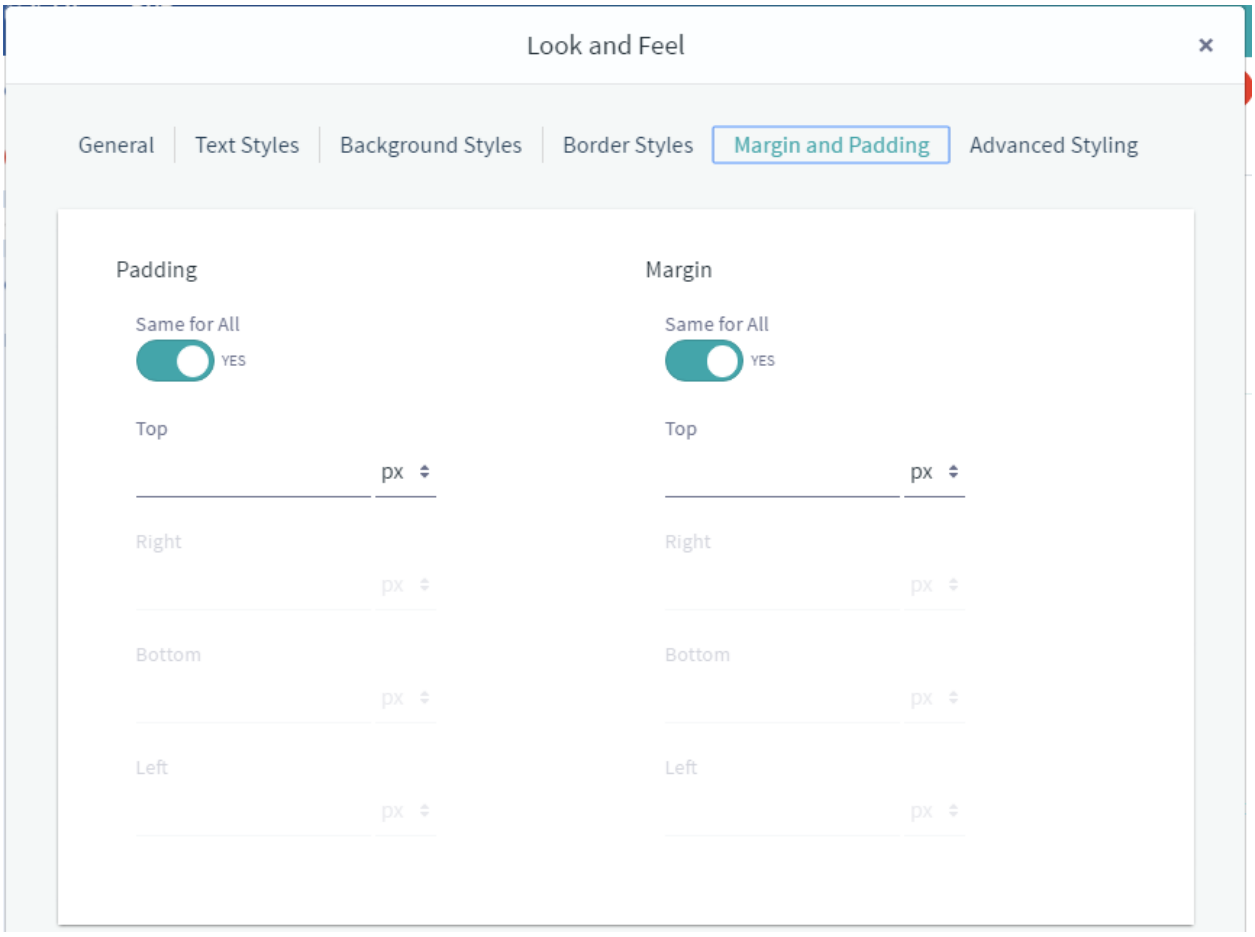


Figure 13.6: The Margin and Padding tab allows you to specify margin and padding lengths for the sides of your app.

Content box, more options appear, allowing you to choose specific kinds of metadata to include. For example, if you have a wiki page with referenced content that you don't wish to include, you can simply check the *Wiki Pages* checkbox and uncheck the *Referenced Content* checkbox. Another option in the Content section of the Export/Import window is the selection of content types. Two familiar content types in your Liferay instance is *Comments* and *Ratings*. If you wish to include these entities in your `.lar` file, select *Change* and select them from the checklist.

Lastly, you can choose whether to include permissions for your exported content. The permissions assigned for the exported app window will be included if the *Export Permissions* selector is enabled. After you've exported your app's data, switch to the *Current and Previous* tab to view ongoing export processes and the history of past exports.

To import app data, you can select the LAR using your file explorer or by dragging and dropping the file between the dotted lines. Your LAR file is uploaded and displayed to you for review. Click *Continue*.

Now that you've uploaded and confirmed your LAR file, you're given a similar screen to what you'd be offered during export. Several of these options are covered in great detail in the Importing/Exporting Pages and Content section. There are a couple of additional options, however, available – *Update Data* and *Authorship of the Content*. Here's options and descriptions for each section:

Update Data

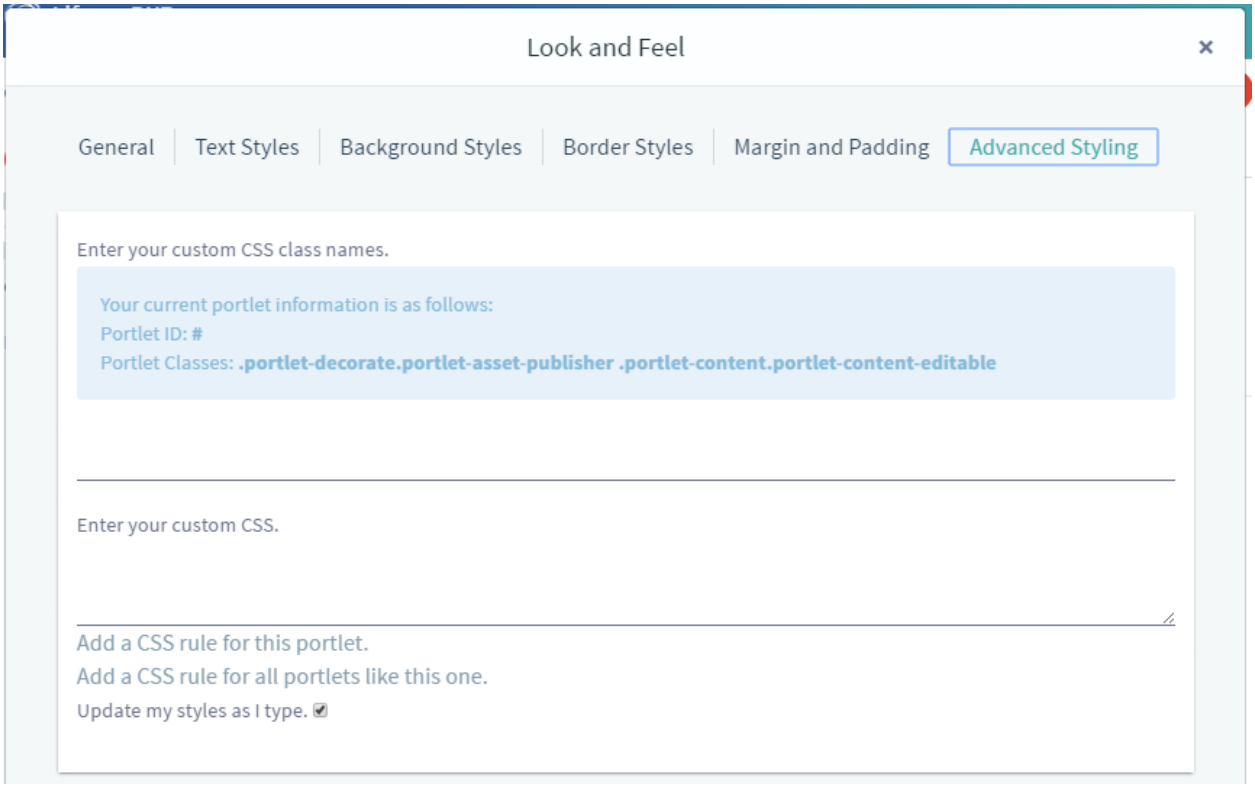


Figure 13.7: The Advanced Styling tab displays your app's Liferay ID and allows you to enter CSS code to customize the look and feel of your app.

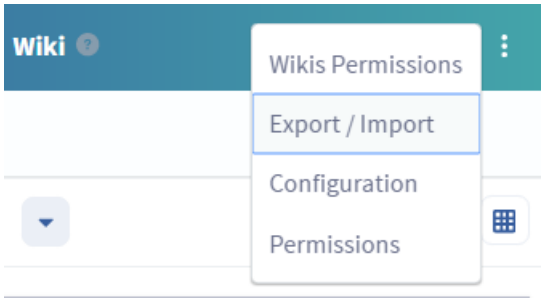


Figure 13.8: You can access the *Export/Import* feature for an app by selecting its Options menu.

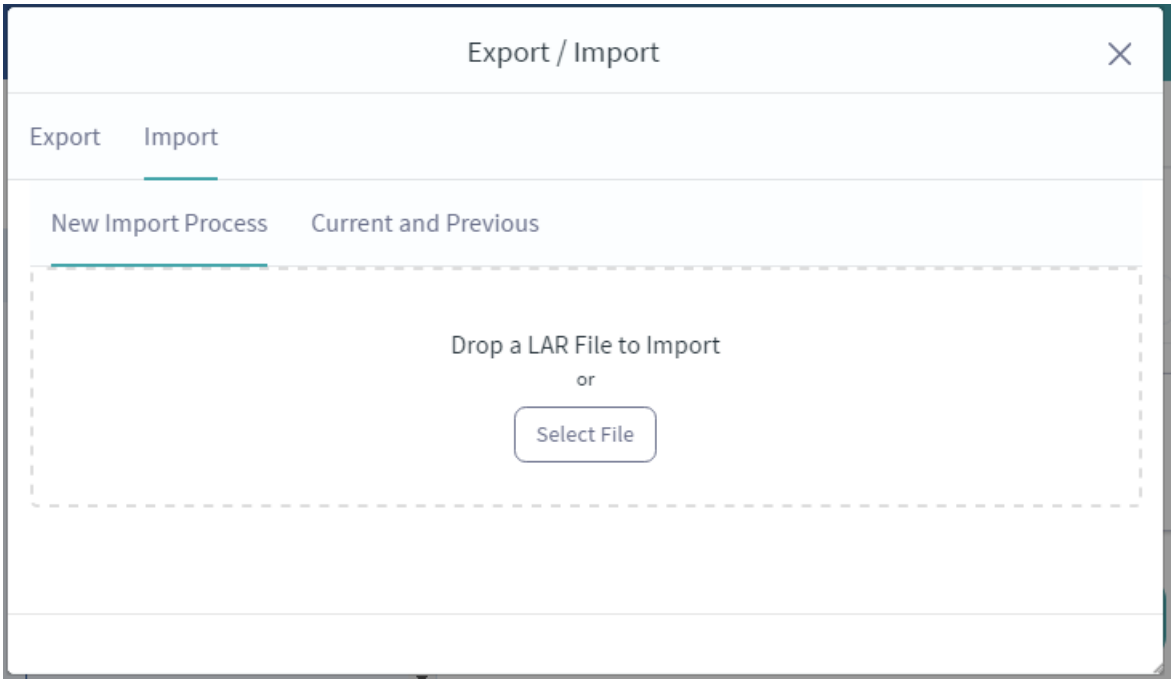


Figure 13.9: When importing app data, you can choose a LAR file using the file explorer or drag and drop the file between the dotted lines.

- *Mirror*: All data and content inside the imported LAR will be created as new the first time while maintaining a reference to the source. Subsequent imports from the same source will update entries instead of creating new entries.
- *Mirror with overwriting*: Same behavior as the mirror strategy, but if a document or an image with the same name is found, it is overwritten.
- *Copy as New*: All data and content inside the imported LAR will be created as new entries within the current site every time the LAR is imported.

Authorship of the Content

- *Use the Original Author*: Keep authorship of imported content whenever possible. Use the current user as author if the original one is not found.
- *Use the Current User as Author*: Assign the current user as the author of all imported content.

Once you've selected the appropriate options, select *Import*, and your app's data is imported and ready for use.

Next, you'll learn how to communicate between applications in a Liferay instance.

13.3 Communication Between Apps

Liferay DXP implements several communication mechanisms across apps including those specified by the JSR-286 standard: public render parameters and events. Public render parameters are easy to use and can be quite powerful. Some Liferay apps provide a configuration UI to help you get the most out of this communication mechanism. To access this UI, open your application's configuration window by clicking on the *Options* icon ( ) and selecting *Configuration*. Then click on the *Communication* tab.

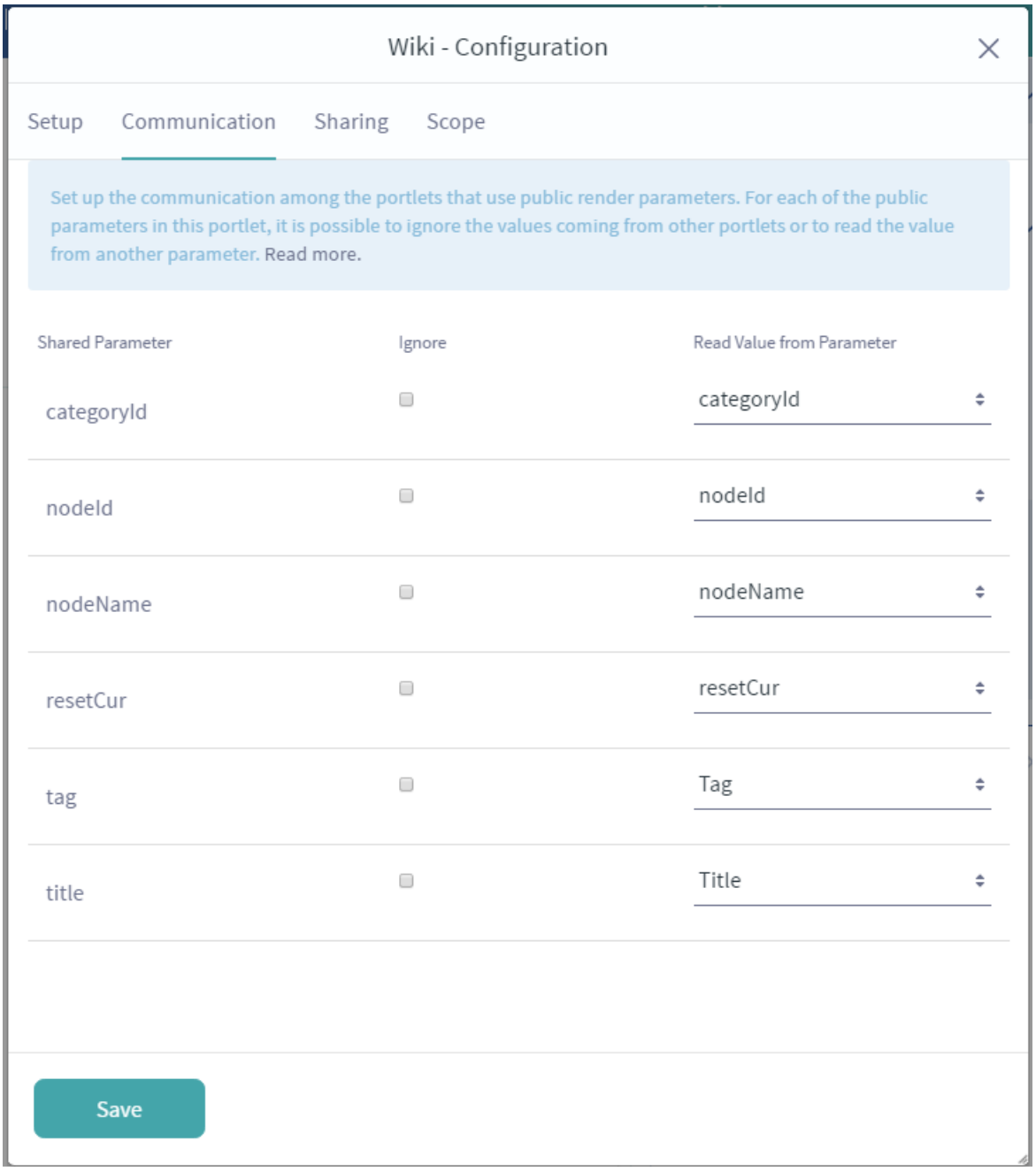


Figure 13.10: You can configure apps to communicate with each other using public render parameters.

The screenshot above is for the Wiki application, which has six public render parameters: `categoryId`, `nodeId`, `nodeName`, `resetCur`, `tag`, and `title`. For each of these parameters, you can configure the app to ignore the values coming from other apps to read the value from another parameter.

Why might it be useful to ignore the values for certain parameters that come from other apps? Consider a common use case for the Wiki application. The Wiki app is often used along with the Tags Navigation app so that when a user clicks on a tag of the latter, the Wiki app shows a list of pages with that tag. In some cases, an administrator may want the Wiki app to always show the front page independently of any tag navigation done through other apps. This can be achieved by checking the *Ignore* checkbox so that the values of the parameter coming from those other apps are ignored.

Reading the value of a parameter from another app is an advanced but very powerful option that allows apps to communicate with each other even if their developers didn't intend them to. For example, imagine that the Wiki application is being used to publish information about certain countries. Imagine further that there's another app that allows browsing countries for administrative reasons. The second app has a public render parameter called *country* with the name of the country. You'd like the Wiki to show the information from the country that's selected in the administration app. This can be achieved by setting the value of the title parameter of the Wiki app to be read from the country parameter of the administration app. Cool, isn't it?

13.4 Sharing Applications with Other Sites

The web was once thought of as a number of islands of applications in a vast universe of "cyberspace." Many web sites attempted to make their island the biggest. Some succeeded to a large extent and some failed. More recently, the concept of the web as an application itself has taken hold, and so widgets have become very popular nowadays. This concept is part of the "Web 2.0" concept and is very much enabled by widgets. What is a widget? A widget is a small piece of code which provides a piece of functionality, can be included on any web site, but does not necessarily have to be hosted by that web site. If you have ever embedded a YouTube video on your own web site so that users could watch a video without actually having to visit www.youtube.com, then you've already used a widget.

Liferay DXP supports serving its apps as widgets. You can embed a particular instance of an application running on your site into another site, such as Facebook. This opens up a whole new avenue of exposure to your web site that you would not have had otherwise. In fact, this is how all those Facebook games work.

To share one of your apps as a widget, open the *Configuration* dialog box from the app's *Options* icon ( ) and select the *Sharing* tab. There are five sub-tabs under sharing: Any Website, Facebook, OpenSocial Gadget, and Netvibes.

Any Web Site

Copy and paste the provided snippet of JavaScript code into the web site to which you want to add the application as a widget. That's all you need to do. When a user loads the page on the other website, the code will pull the relevant app from your site and display it.

Facebook

You can add any Liferay app as an application on Facebook. To do this, you must first get a developer key. A link for doing this is provided to you in the Facebook tab. You'll have to create the application on Facebook and get the API key and canvas page URL from Facebook. Once you've done this, you can copy and paste their values into the Liferay's Facebook tab. Save the configuration and navigate back to the Facebook tab in Liferay. You're given the Callback URL, which you can copy and paste into Facebook. By doing this, when

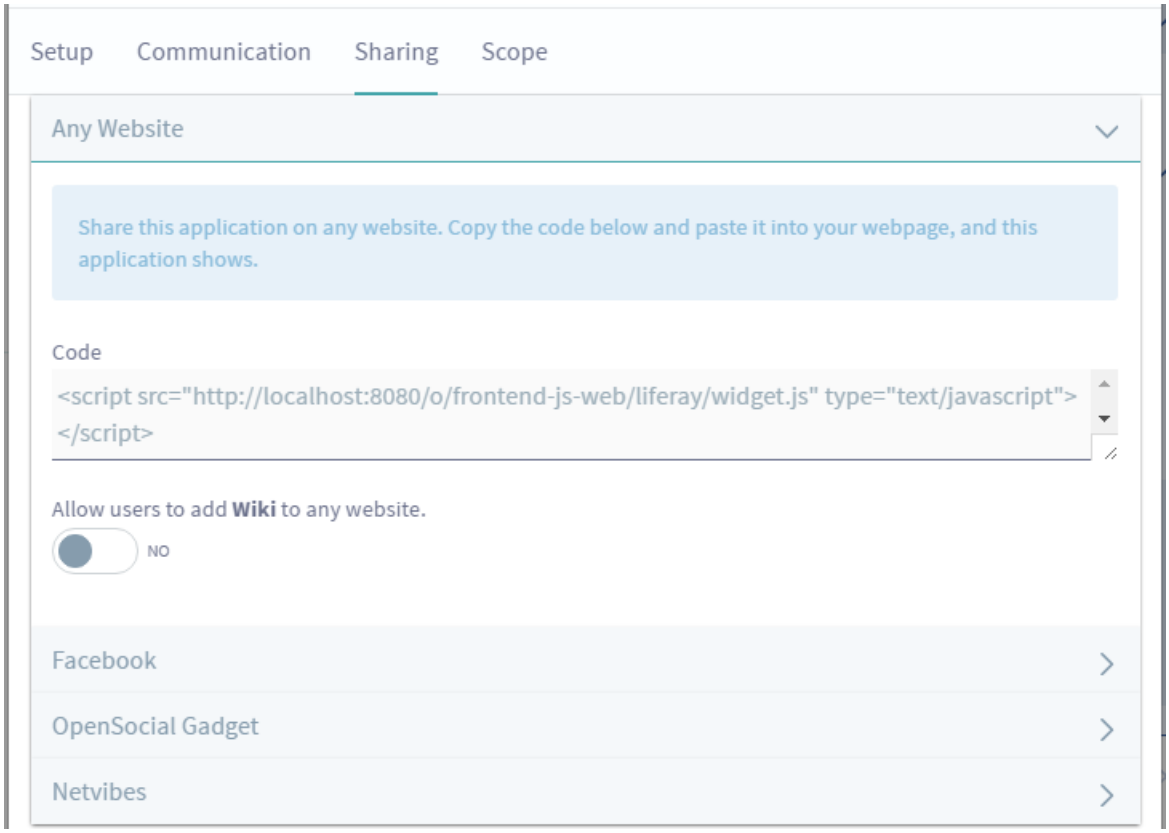


Figure 13.11: The Sharing tab in your app's Configuration menu lets you share your app in a variety of ways.

opening your app in Facebook, the correct callback URL is used to render the application. You can also enable the *Allow users to add [application-name] to Facebook*. Then you can navigate to your app's Options menu and select *Add to Facebook*.

By having the ability to share your apps, this makes Liferay DXP a fantastic platform upon which to build applications for Facebook.

OpenSocial Gadget

OpenSocial comprises a container and a set of APIs for social networking and other web applications. iGoogle is a service provided by Google that lets users create a customizable page and add *Gadgets* to that page. Liferay DXP can serve up applications to be used as OpenSocial Gadgets on any OpenSocial-compatible pages.

To serve a Liferay app on an OpenSocial platform, copy and paste the provided gadget URL and add it to the appropriate configuration page of the OpenSocial platform you're using. Your Liferay instance will serve that app directly onto that platform's page. The URL provided is unique to the specific instance of the app, so you could serve multiple instances of the same app as different OpenSocial Gadgets.

You could use this feature to allow users to view what's happening on your Liferay instance at a glance, using Asset Publishers or custom RSS feeds. You could also use Liferay's API to build your own app and provide the URL for users to place on their OpenSocial pages.

From the Sharing tab in the Configuration menu, you can also enable the selector *Allow users to add [application-name] to an OpenSocial platform*. Click *Save* and revisit the *Options* button of your application in Liferay DXP. You'll notice a new button named *Add to an OpenSocial Platform*. When selecting this new button, the URL is provided to you for sharing the application to an OpenSocial platform. This option provides a

shortcut for obtaining the URL and also gives a way for users to copy the URL that wouldn't have permission to access the app's Configuration menu.

Netvibes

Netvibes offers a similar service to iGoogle—users can log in, create their own personal portal, called a *dashboard*, and add customizable widgets to the dashboard that they create. To set up Netvibes support for a particular app, enable the *Allow users to add [application-name] to Netvibes pages* selector. You can then use the provided URL to create a custom Netvibes widget based on the instance of the app that you're using.

Next, you'll learn how to set permissions for Liferay applications.

13.5 Application Permissions

All of Liferay's apps support Liferay DXP's robust, fine-grained permissions system. Some higher level permissions can be configured in the permissions tab of the app's configuration dialog box. You can grant roles permission to:

- Add a display template
- Add the app to a page
- Configure the app
- Modify the app's permissions
- Modify the app's preferences
- View the app

To set these permissions, go to the app's *Options* icon ( ) and click select *Permissions*. This shows you a table of roles defined in the Liferay instance. Use the check boxes to grant certain permissions to different roles. Click *Save* after you've made your selections.

Beyond this, specific permissions are generally defined for specific applications. For example, the Message Boards application contains a *Ban User* permission. This makes no sense in the context of another app, say, the Blogs application. You'll go over permissions for specific applications in the sections for those applications. Next, you'll explore how to sharing applications.

13.6 Application Scope

As you learned earlier, roles can be scoped by the instance, by a site, or by an organization. A role only takes effect within its scope. For example, a Message Boards Administrator role with complete access to the Message Boards app has different permissions based on the role's scope. If it's a instance role, members have permission to administer message boards across the entire instance. If it's a site role, members only have permission to administer message boards within the site where they've been assigned the role. For organizations with sites, site roles are automatically assigned to organization members based on the organization roles they have. So for an organization-scoped Message Boards administrator role, members only have permission to administer message boards within the site of the organization that assigned the role to them.

You've also heard the word *scope* refer to the data set of an app. By default, when an app is added to a page in a site, it is *scoped* for that site. This means that its data belongs to that site. If the app is added to a page in a different site, it employs a completely different data set. This enables you to place a Message Boards app in one site with one set of categories and threads, and place another Message Boards app in different site with a different set of categories and threads.

Scoping by site means that you can only have one Message Boards app per site. If you add one Message Boards app to a page in a site and add another Message Boards app to a different page in the same site, the second Message Boards app contains exactly the same data as the first. This is because, by default, the Message Boards app is scoped by site. Most of Liferay DXP's other apps also default to being scoped by site.

To avoid this limitation, many Liferay apps can be scoped by page. In this case, the data sets of page-scoped apps serve a single page, not an entire site. If you set the scope of an app to *page* instead of *site*, you can add any number of these apps to different pages, and then they have different sets of data. This allows you to have more than one message board per site if you wish. Most apps, however, default to the "native" configuration, and have their scopes set to the site where they are placed.

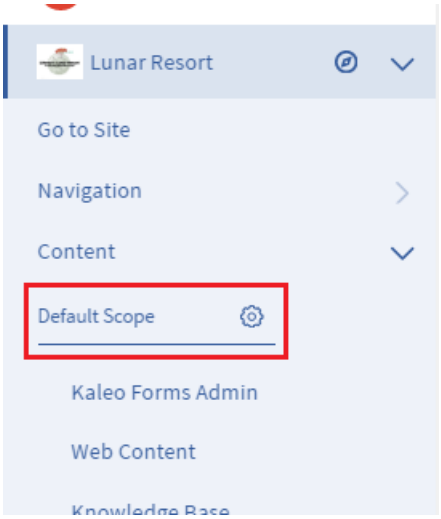


Figure 13.12: When defining a page scope for an app, the Menu provides a *Default Scope* dropdown.

Unless otherwise noted, all the apps in this chapter support scoping by instance (global), site (default), or page . This grants you some flexibility in how you want to set up your Liferay instance. You can configure the scope of an app with just a few simple steps.

1. Click the *Options* icon ( ) in the app window.
2. Select *Configuration*.
3. Select the *Scope* tab.
4. Use the drop-down menu to set the scope.

Once you've defined a page scope for an application, the Menu provides a *Default Scope* dropdown that allows you to select the page you want to create content on.

That's all it takes to change the scope for a particular application instance. By setting the scope to the current page, you can add as many of these apps to a site as you want, provided they are all added to separate pages.

Another useful feature of Liferay's apps is Configuration Templates.

13.7 Configuration Templates

Once you've configured an application, Configuration Templates enable you to save those settings in a reusable template. If someone goes in and changes the settings of a particular app, it then becomes easy to

Figure 13.13: You can change the scope of your application by navigating to its Configuration menu.

revert those changes back to the original configuration template. Configuration templates are only available for apps placed on a page. Applications available from the Menu do not provide configuration templates.

To create a configuration template, click the *Options* icon ( ) from the menu in the app's title bar and select *Configuration Templates*. If the current settings of the app you're configuring are the ones you want to save, click the *Save Current Configuration as Template* button. If not, change the settings of your app until it's configured the way you want it, and then click the button.

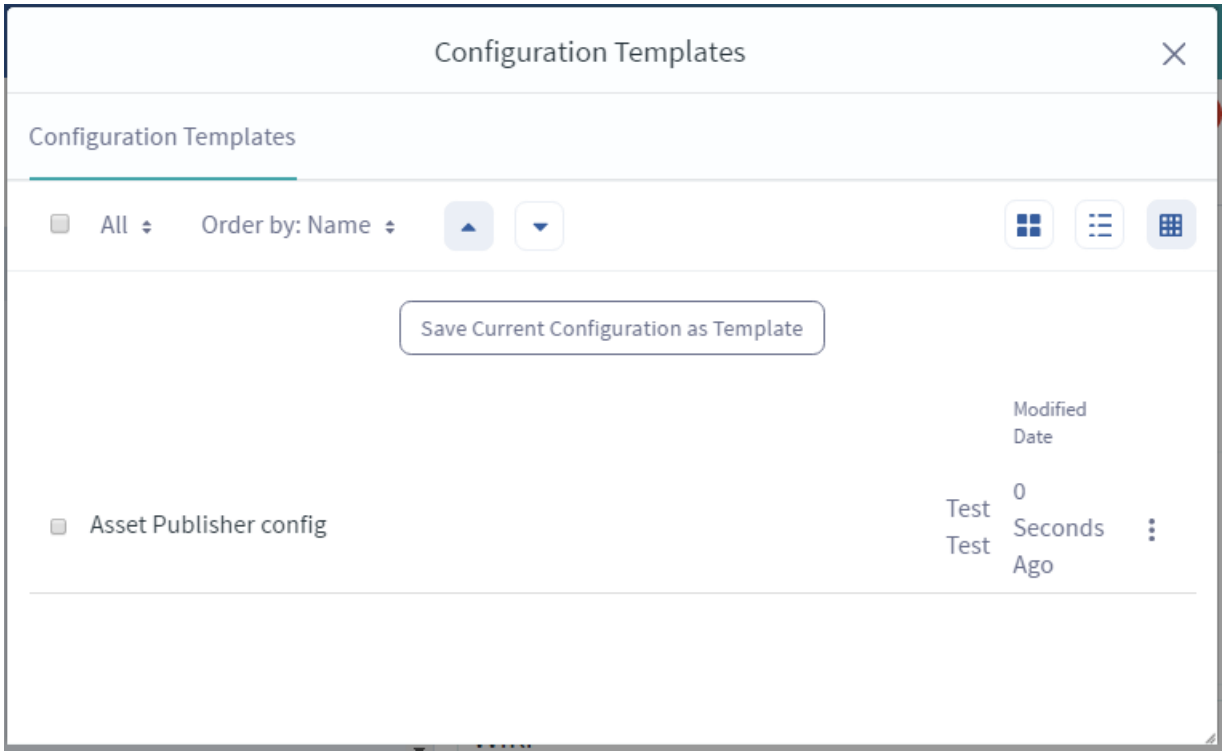


Figure 13.14: Create a configuration template to save your app's configuration settings.

There is only one field to fill out. Enter a name for your template and click *Save*. You should now see your configuration in the list. If you ever need to revert the app to these archived settings, you can click *Actions* ( ) → *Apply* next to the configuration template you want to apply.

Unless otherwise noted, all of the apps in Liferay DXP support this feature. This is particularly useful for

apps that have a lot of configuration options, such as the Message Boards application.

In this chapter, you've explored the configuration options available for Liferay applications. You learned how to customize your applications, export/import app data, communicate between apps, take advantage of different scopes, and save your app's configuration settings. You also examined the different uses of social applications like Facebook and Netvibes for your Liferay applications. In all, Liferay DXP gives you an abundance of options to leverage the full capability of your applications.

13.8 Styling Apps with Application Display Templates

The application display template (ADT) framework allows Liferay administrators to override the default display templates, removing limitations to the way your site's content is displayed. With ADTs, you can define custom display templates used to render asset-centric applications. For example, you may want to show blog entries horizontally instead of vertically, or list your assets in the asset publisher application in different sizes.

You'll go through a simple use case to illustrate how creating a custom ADT can improve your site. Suppose you're customizing the Lunar Resort site and want to allow users to communicate with other interested travelers. Specifically, you want to configure the Wiki app for collaboration with Facebook or Twitter. With ADTs, you can launch a template editor, create a custom template, and configure your app host that template. Custom templates let you re-skin your application and give you ultimate control over its appearance and functionality in your Liferay instance.

Before attempting to change the ADT for your application, you'll need to select a site for your custom template to reside in. Choosing the *Global* context makes your template available across all sites. To choose a site to house your ADT, navigate to the Site Administration dropdown list by selecting the *Site Selector* button ( ). If you select the Global context, the *Application Display Templates* page of Site Administration's Configuration Menu shows you a list of sample templates available for your apps. These sample templates differ from the default templates already configured in the apps. If you choose a site to host your template, you must create a custom template for that site's apps.

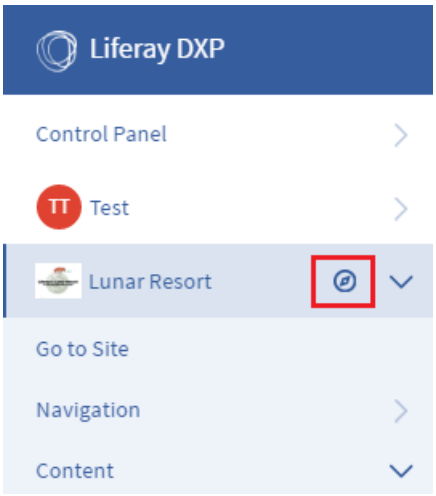


Figure 13.15: The Site Administration dropdown menu lets you choose the context in which your application display template resides.

If you'd like to add an existing ADT, select the template from the list. The list below specifies the apps that can be customized using ADTs.

- *Asset Publisher*

- *Blogs*
- *Breadcrumb*
- *Categories Navigation*
- *Documents and Media*
- *Language Selector*
- *Navigation Menu*
- *RSS Publisher*
- *Site Map*
- *Tags Navigation*
- *Wiki*

To create a new ADT, click *Add* and select the template you'd like to create, based on application type. Then enter the name and, optionally, a description and a small image to use. You can select the language type for your template (FTL or VM). Lastly, the *Script* option lets you browse your file system for a template on your file system, or you can use the editor and create one directly. On the left side of the template editor, you'll notice a palette of common variables used for making templates. This is a great reference when creating your template. To place one of the variables into the template editor, simply position your cursor where you want it placed, and click the variable name.

Think it can't get any better? Guess again! If the variable name doesn't give you enough information on the variable's functionality, you can hover your pointer over it for a more detailed description. Because there are multiple kinds of ADTs, there are also different variables for each ADT. Thus, each template has a different set of variables only applicable for that specific template.

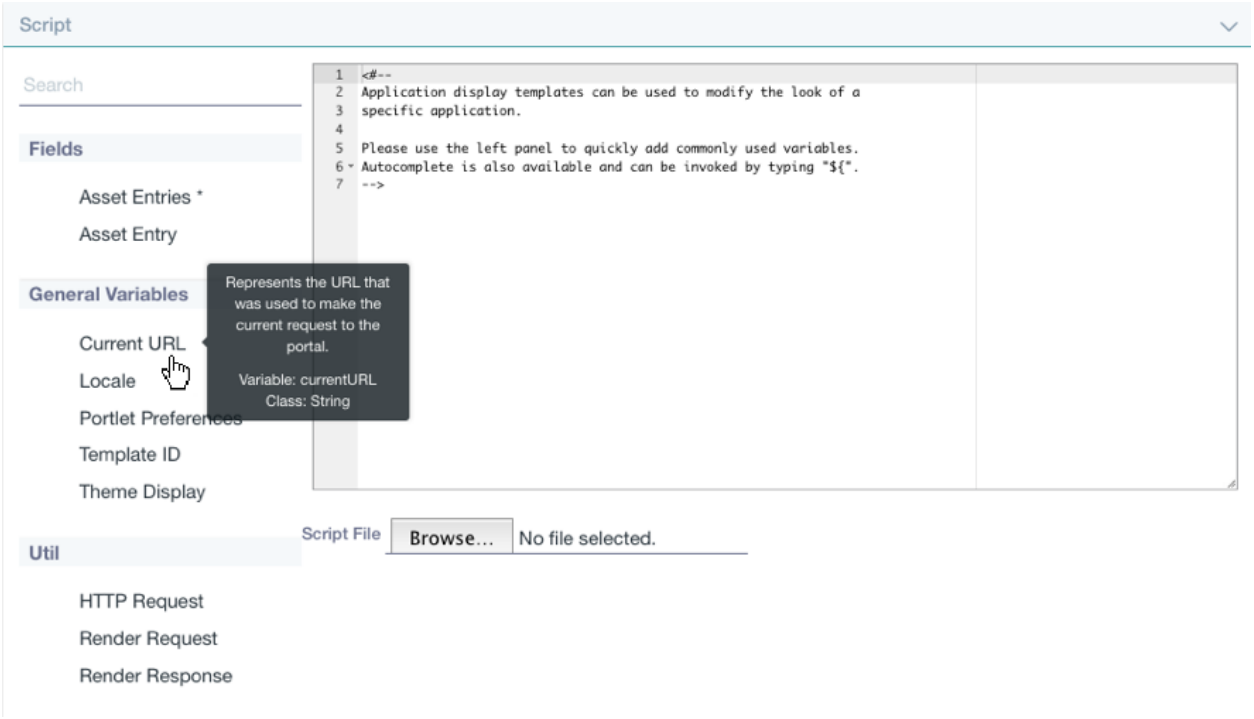


Figure 13.16: Liferay offers a versatile script editor to customize your ADT.

You can also use the autocomplete feature to add variables to your template. It can be invoked by typing *${* which opens a drop-down menu of available variables. By clicking one of the variables, the editor inserts

the variable into the editor.

You also have the ability to embed same-type templates into other templates. For example, suppose you have an existing Wiki ADT and would like to create another similar Wiki ADT. Instead of starting from scratch, you can import the existing Wiki ADT into your new one and build off of it. In other words, you can utilize ADTs as generic templates which allow for reusable code to be imported by Velocity or FreeMarker templates in the system. For more information on how to create a custom template, visit the Implementing Application Display Templates tutorial.

After you've completed the initial set up and saved your ADT, you can manage your ADT through its *Actions* ( ) button. This provides several options:

- *Edit*: lets you modify the ADT's setup properties.
- *Permissions*: lets you manage the permissions *Update*, *Permissions*, *Delete*, and *View* for the ADT.
- *Copy*: creates a copy of the ADT.
- *Delete*: deletes the ADT.

Additionally, your ADT generates a static URL and a WebDAV URL. These values access the XML source of your template. You can find these URLs by clicking the ADT from the menu and expanding the *Details* section. With the WebDAV URL, site administrators are capable of adding, browsing, editing, and deleting ADTs on a remote server. If you'd like to learn more about what the WebDAV URL can do, visit the section on WebDAV access.

Note: Embedding apps into ADTs, although possible, is not recommended because this could cause conflicts with other apps or unexpected behavior (e.g., embedding an app that aggregates data to the breadcrumb). If embedding an app into an ADT is your only option, make sure it does not interfere with other apps.

To enable your ADT for an application, navigate to the app you want to modify and open its *Configuration* menu. In the *Setup* tab, select your ADT from the *Display Template* drop-down menu. Also, you can manage site-specific display templates for your app: do this by clicking the *Manage Display Templates for [SPECIFIC_SITE]* link next to the *Display Template* drop-down menu. A window will display with a list of your configured templates only available for your site with options to add new templates or edit existing templates.

Now that you know the general functions of ADTs, you'll create your own. This brief demonstration will show you just how easy, yet powerful, ADTs can be for your Liferay instance.

1. Add the Media Gallery application to a page by navigating to *Add* ( ) → *Applications* → *Content Management* → *Media Gallery*.

2. Click the app's *Add* button ( ) → *Multiple Media* and select two custom photos to display. Then click *Save*, and navigate back to the main application screen.

3. Notice the default format of the pictures. To change the display template for this app, navigate to *Options* ( ) → *Configuration*.

4. From the *Display Template* drop-down menu, select *Carousel*. Then click *Save*.

   The Media Gallery application is transformed into a carousel slideshow. At this time, it's perfectly natural to be experiencing "I can conquer the world" feelings, just as Liferay's mascot, Ray, exudes in the image above. ADTs have that kind of power to transform your site into an enjoyable and convenient home for users.

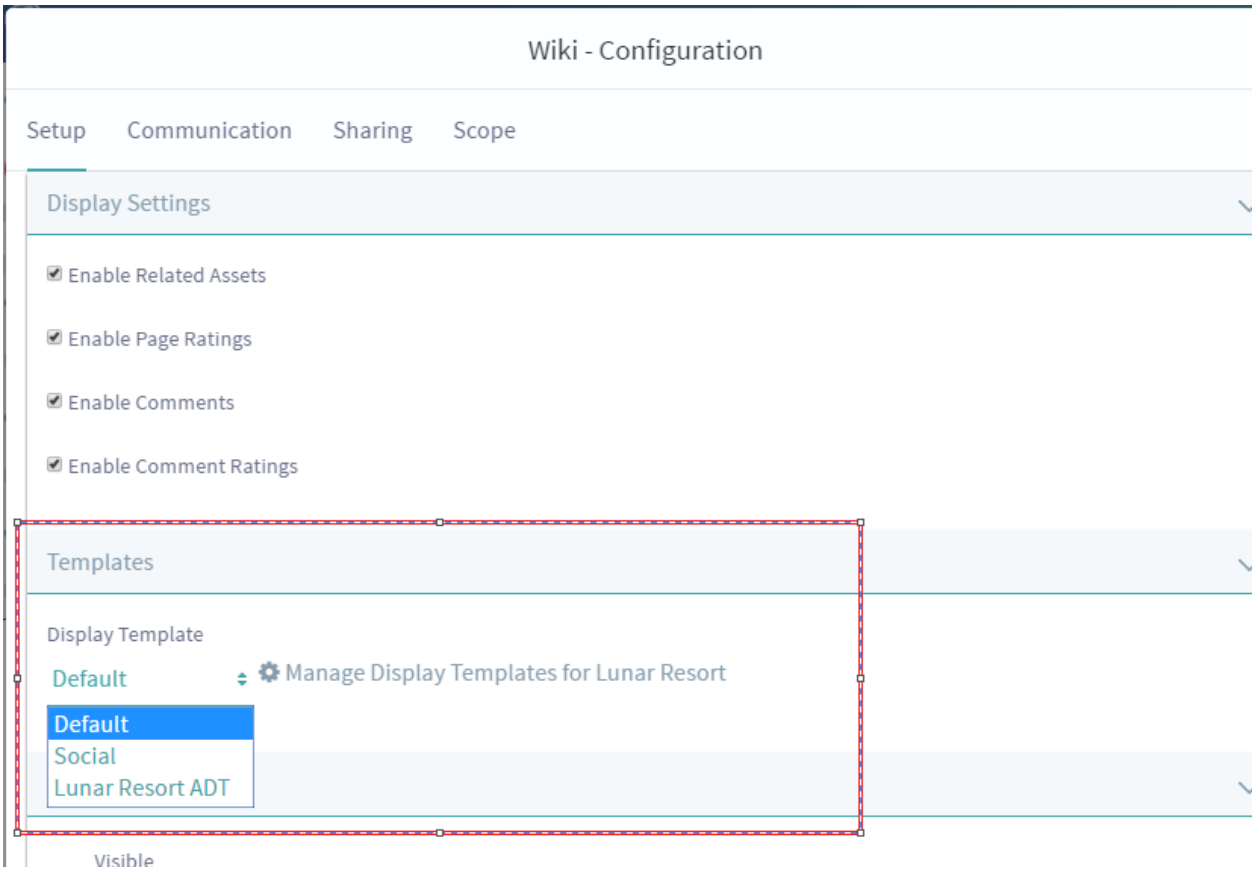


Figure 13.17: In the *Configuration* menu of an app, you can edit and manage available ADTs.

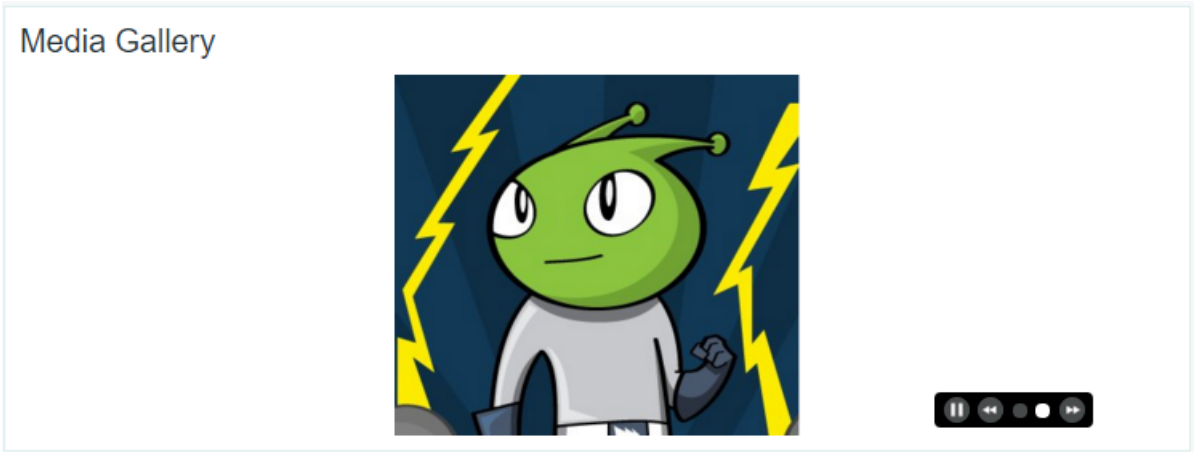


Figure 13.18: After applying the Carousel ADT, your pictures are displayed as a carousel slideshow.

Customizing the user interface of Liferay DXP's bundled apps provides the ultimate customization experience for Liferay users.

Chapter 14

Collaboration

What comes to mind when you think of *collaboration*?

George Lucas, Steven Spielberg, and John Williams collaborated to give us the thrilling adventure story, fantastic stunt sequences, and adrenaline pumping score in the Indiana Jones films.

The United States, Russia, and other nations combined efforts to launch and maintain the International Space Station to conduct biological and physical research that may allow us to explore Mars, Pluto, and beyond.

Bono, Bill Gates, and others join forces annually at the World Economic Forum to take on world issues such as poverty and AIDS.

Now, you might not be a celebrity or a renowned scientist, but you have important things to accomplish. And no matter whether you're saving lives, exploring the universe, or entertaining the masses, you're typically much more effective doing it with others. As you work together, your collective ability to brainstorm ideas, produce something unique, and present it to the outside world are key components to successful collaboration.

Liferay's collaboration suite resonates with apps and features that foster excellent communication. Its Message Boards app gives your team a platform for discussions, questions and answers, and comments. Blogs publish your ideas using rich content, so readers can understand them clearly and respond to them. Collaboration is enhanced in all these applications through *mentioning* other users–tagging them by name to get their attention or give them kudos.

As you produce digital assets–documents, videos, audio–you can store and share them using the suite's Documents and Media Library. It both provides its own repository and connects to external CMIS repositories and even Google Docs™ via the Liferay Plugin for Google Drive™ from the Marketplace.

Finally, you can collaborate on and publish content in a number of different ways. You and your colleagues can build a wealth of information using Wikis and create hierarchies of official documentation in a Knowledge Base.

These are just some of the ways the Collaboration Suite helps you communicate, produce, and present. In short, it empowers you to labor together with others in doing amazing things!

Chapter 15

Managing Documents and Media

Liferay DXP's Documents and Media library provides a mechanism for storing files online using the same type of structure that you use to store files locally. You can use it to store files of any kind; it serves as a virtual shared drive, and can mount and browse external repositories. Its companion app, the Media Gallery, displays selected content from the Documents and Media library. It can display image, audio, and video files. Other features in the Documents and Media library include customizable document types and metadata sets, automatic document preview generation, and support for mounting multiple external repositories.

Liferay Sync synchronizes your local copies of the Documents and Media libraries with your site's libraries. It lets you manage files from your desktop machine and mobile device. It provides the best way to manage your files on all your devices.

Let's get started with Documents and Media by exploring how to publish files.

Chapter 16

Publishing Files

As you create sites, you'll probably want to share files on them. Liferay DXP's Documents and Media library (Document Library) let's you upload and publish all kinds of files on your sites. Pictures, videos, spreadsheets, slide presentations and more can be stored in and shared from the Document Library. Document Library instances can be scoped to a Liferay DXP instance, site, or page, so you can work with files where they're relevant.

In these guides, you'll learn how to add files, display them, and collaborate on them. You can upload all kinds of files, including custom file types, to the Document Library. Users can then browse and view files from the Media Gallery or Documents and Media apps. You'll learn how to use both apps. And lastly, you'll learn how to collaborate on files from within several environments, including your browser, local desktop, and mobile device. We've pulled out all the stops to simplify online file management. Let's get the ball rolling by adding files to a Document Library.

16.1 Adding Files to a Document Library

There are several reasons why you might want to upload files to your site. You might want to share photos and video files or embed them in web content. Or you might simply want to back them up from your device to your site. You may have important documents that you want to share with other users. And you might want to collaborate on documents with other users. All of these examples include first uploading a files to your site.

To demonstrate adding files to a Document Library, we'll implement a use case for the Lunar Resort. Resort photographers take lots of photos and video footage of resort guests having the time of their lives. As a courtesy to the guests the staff stores each party's photos and videos on the resort site and shares them with the parties. At first, you might think "How does a resort's sharing of pictures and videos relate to me and my site?" Well, even if your company isn't in the tourism business, it probably has similar requirements for managing and publishing files. One common requirement is the ability to enable specific groups of users to store, organize, and share files. And no matter what kinds of files you work with, the fundamentals of granting file operation permissions is the same. Let's explore how to do this for the Lunar Resort.

Granting File Permissions and Roles

The Lunar Resort needs to manage carefully who can add, view, and update groups of files. The resort stores in their Documents and Media libraries all kinds of files for various purposes. The libraries not only

contain guest media but also departmental documents and media. Most of the documents are meant to be disseminated within the departments only. Using Liferay DXP, we can implement the resort's use cases. You see, Liferay DXP's Roles and Permissions let you precisely specify access to documents and media files. In addition, the Document Library's folder permissions help you organize media. Let's get started on the right foot with the Lunar Resorts guest media files by leveraging roles, permissions, and folders in Documents and Media.

Since the Lunar Resort's *Souvenir and Memorabilia* staff members must be able to upload and organize guest media, let's enable a role for these purposes. Sign in as the Liferay DXP admin and open the *Menu* ( ). Then navigate to *Control Panel* → *Users* → *Roles*. If the staff group is a part of an Organization, and that organization has a site, create an Organization Role. Otherwise, create it as a Site Role. Name the role *Resort Guest Media Manager*. Once you've created it, click the role's Actions icon ( ) and select *Define Permissions*. A screen appears for you to define the role's permissions.

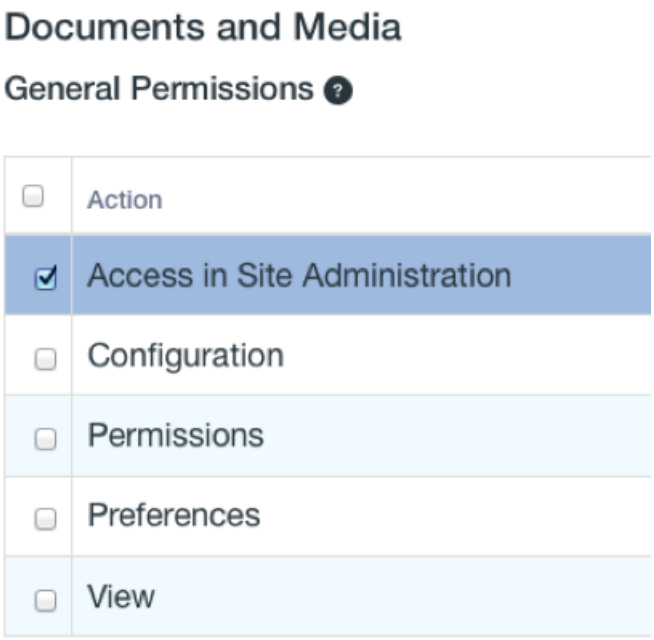


Figure 16.1: It's often helpful to define a role for specific users to access Documents and Media from Site Administration.

In the role's permission definition screen, navigate to *Site Administration* → *Content* → *Documents and Media*. In the *General Permissions* section, select *Access in Site Administration* and click *Save*. This role will manage your special set of media files. Once you've created the role, assign it to the users who manage the media. Snappy Fingers (username: *snappy*) is the Lunar Resort's trigger-happy photographer. We assigned her to the *Resort Guest Media Manager* role.

Note: To learn how to create a role and define a role's permissions, refer to Roles and Permissions.

You've created your specialized role and assigned users to it. In the Documents and Media library, you need a place for that role to manage Lunar Resort guest media. A Documents and Media folder fits the bill. In the folder we'll create, the role will add, organize, and share the guest media files.

Using the Add Menu

Open the Menu, click on your site's name, and navigate to *Content → Documents and Media* for your site. The Documents and Media screen appears and displays the Documents and Media library's *Home* (its root folder). As you add files and folders to the document library, they're listed here.

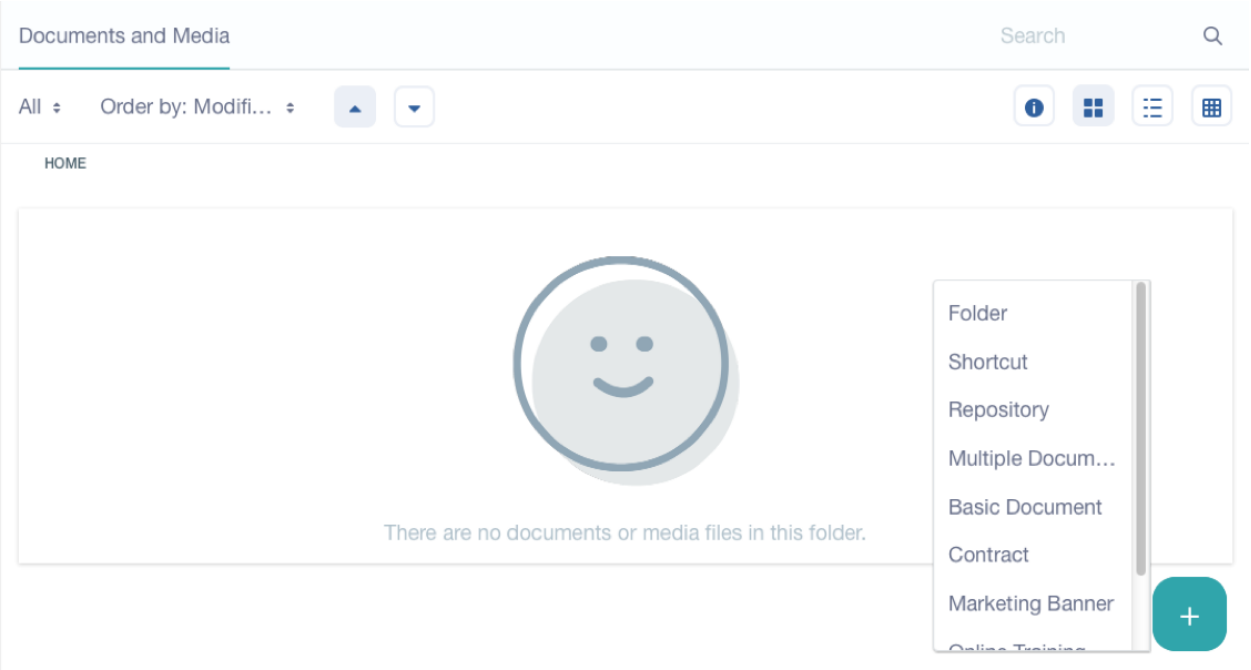


Figure 16.2: The Documents and Media's *Home* folder starts empty. But the Add menu lets you upload and add all kinds of documents to the library.

Click the *Add* icon ( ) to show what you can add to the Document Library. You can add documents, folders, and shortcuts the same as you do on your local file system. You can even configure access to an entirely different repository. Here are the Add menu's options:

Folder: lets you create a new location in the app's file system.

Shortcut: allows you to create a shortcut to any document that you can view. You can set permissions on the shortcut to specify who can access the original document through the shortcut.

Repository: allows you to add access to an external repository. Refer to Repostory Types to learn how to add access to them.

Multiple Documents: allows you to upload several documents at once.

Basic Document: allows you upload a single file that you would like the default document type, "Basic Document," to apply to. By default, basic documents are not described by any metadata sets.

Google Docs: available through an additional Liferay Plugin for Google Drive™ from the Marketplace. This option lets you create a file entry that links to a Google document. The section Accessing Google Docs™ explains how to use this feature.

The remaining items in the Add menu are default document types that are each described by a unique metadata set. When you add a document belonging to a specific document type, a form appears for picking the file to upload and entering the data defined by the document type's metadata set. The *Contract* type appears below as an example.

Contract: lets you upload a file that you would like the "Contract" document type to apply to. This document type is intended to be used to describe legal contracts. By default, contracts are described by

effective date, expiration date, contract type, status, legal reviewer, signing authority and deal name fields.

Any custom documents types that have been defined also appear in the Add menu. If a document type has been created that matches the document you would like to upload, you can select that document type from the Add menu. The Document Library applies the metadata fields associated with the document type to your document. On selecting a document type, you're prompted to fill in the fields associated with it.

Creating Folders

Let's continue with our example and create folders to organize the Lunar Resort guest photos and videos that we'll upload. From the document library's Add menu, select *Folder*. The new folder form appears. Since the folder is for storing Lunar Resort guest media, name the folder *Resort Guest Media*. You can optionally describe the folder. Initially, anyone can view the folder, but we'll change that after we create it. Expand the *Permissions* section. By default, site members are able to add files, subfolders, and shortcuts and subscribe to changes to the folder's files. Click the *More Options* link and deselect all of these checkboxes, as we only want resort media managers to modify the folder's files. Click *Save* to finish creating the folder.

Note that, document type restrictions and workflow definitions can be associated with each folder. Child folders inherit their parent folder's document type restrictions, by default. If workflow is enabled for the Document Library, you can specify workflow definitions per folder and document type. You can specify a folder's document type restrictions and workflow definitions from the folder's edit screen, after the folder's been created.

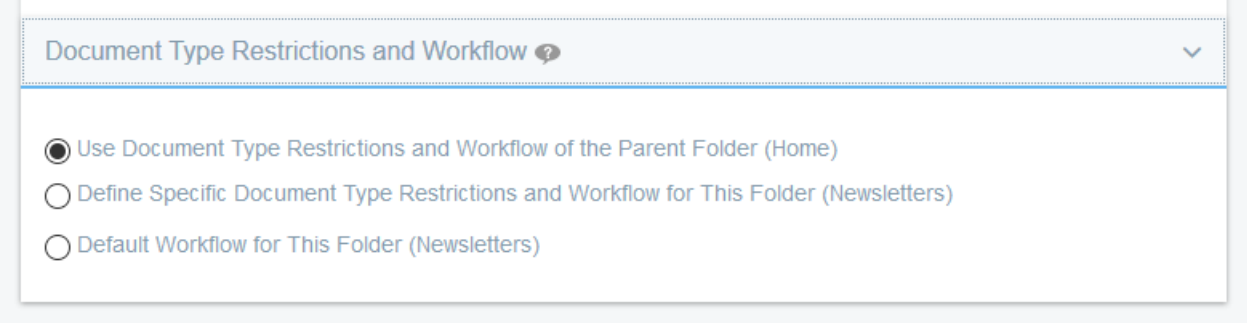


Figure 16.3: From a folder's Edit screen, you can restrict the document types it supports and select the folder's workflow.

Document types are a powerful way to enforce rules for documents. For our example's folder, however, we'll keep the default supported document types and workflow settings.

You've created a folder named *Resort Guest Media*. You can now specify its permissions.

Setting Folder Permissions

Since we only want the *Resort Guest Media Manager* role to manage files in it, we must fine tune the folder's permissions. Click on the folder's *Actions* icon (⋮) and select the *Permissions* action. The Permissions window appears.

The folder permissions enable a role to perform the following actions:

Access: Access the folder's contents from a Windows desktop.

Update: Edit the folder's attributes and move the folder under a new parent folder.

Add Subfolder: Create folders within the folder.

Add Shortcut: Create a shortcut (link) to any Documents and Media file the role is authorized to view.

Subscribe: Receive email notification when files are added to or modified in the folder.

Add Document: Add a new file to the folder.
Permissions: View and modify the folder's permissions.
Delete: Move the folder to the Recycle Bin.
View: View the folder.

Grant all the permissions to the *Resort Guest Media Manager* role, revoke all permissions from all the other roles, and click *Save*. Users assigned to the *Resort Guest Media Manager* role are now the only non-admin users who can upload and manage media files in the *Resort Guest Media* folder.

You've created the *Resort Guest Media* folder and set appropriate permissions for it.

While you're signed in as the admin user, click on Documents and Media's *Options* icon ( ) and select *Configuration*. From here you can specify the email sender and email templates for email sent to Document Library folder subscribers.

Now, sign in as a *Resort Guest Media Manager* role user. Since Snappy, the Lunar Resort's photographer, belongs to this role, you can sign in as her. Then go to *Documents and Media* in Site Administration, navigate to the *Resort Guest Media* folder, and click the folder's name. The folder provides a workspace for members of the *Resort Guest Media Manager* role to manage files and subfolders.

Snappy has plenty of pictures of guests Mr. and Mrs. Spacey on her local file system. To separate the Spacey's files from those of other guests, we'll create a subfolder named *Spacey Party*. To create it, click the Add icon, and select the *Subfolder* action. In the *New Folder* form that appears, name the folder *Spacey Party* and describe the folder's purpose (e.g., "The Spacey's photos from their visit to the Lunar Resort"). Click the *More Options* link and deselect all the permissions within it, as we only want media managers to be able to edit the folder. *Save* the new subfolder. You return to the folder *Resort Guest Media*, which now lists your new subfolder *Spacey Party*. Navigate into it. The Document and Media library's navigation breadcrumbs show your folder's path: *Home → Resort Guest Media →Spacey Party*.

In your Document Library's *Resort Guest Media* folder, you've created a subfolder named *Spacey Party*. Next, let's add the Spacey's photos to the subfolder.

Adding Multiple Documents

There are a few different ways you can add image files, or any file for that matter. You can add them one at a time or add multiple files simultaneously. Let's add all the images at once. Click the Add icon and select *Multiple Documents*. The *Add Multiple Documents* screen appears and displays an area for dropping files in. You can drag and drop files into the area, or you can browse for and select multiple files by clicking the *Select Files* button. Use the method that's easiest for you. As you drop in files or select files, the Add Multiple Documents screen lists them. On the side of the screen, the *All Selected Files* section lists several options for the files. You can fill in a common *Description* for the files. You can also specify a *Document Type* to apply. Since we're uploading image files, we'll use the *Basic Document* type, which is the default type. There are also options for categorizing and tagging the selected files, and assigning them default permissions. We'll tighten up permissions by clicking the *More Options* link and deselecting all the permissions. When you're ready to upload the files, click *Publish*. The Add Multiple Documents screen stays active, ready for you to add more files. When you're done adding files, click the Back icon ( ) at the top of the screen. You're taken back to the folder you're adding files to.

Using the Documents and Media Management Bar

Documents and Media lists your current folder's subfolders and file entries. A *file entry* is the Document Library's representation of a file. It wraps the file to better leverage it within Liferay DXP and to associate additional information and metadata with the file. File entries are displayed using icons, by default. You can select either icon ( ), descriptive ( ), or list ( ) display style. The icon display style shows a file as a cover

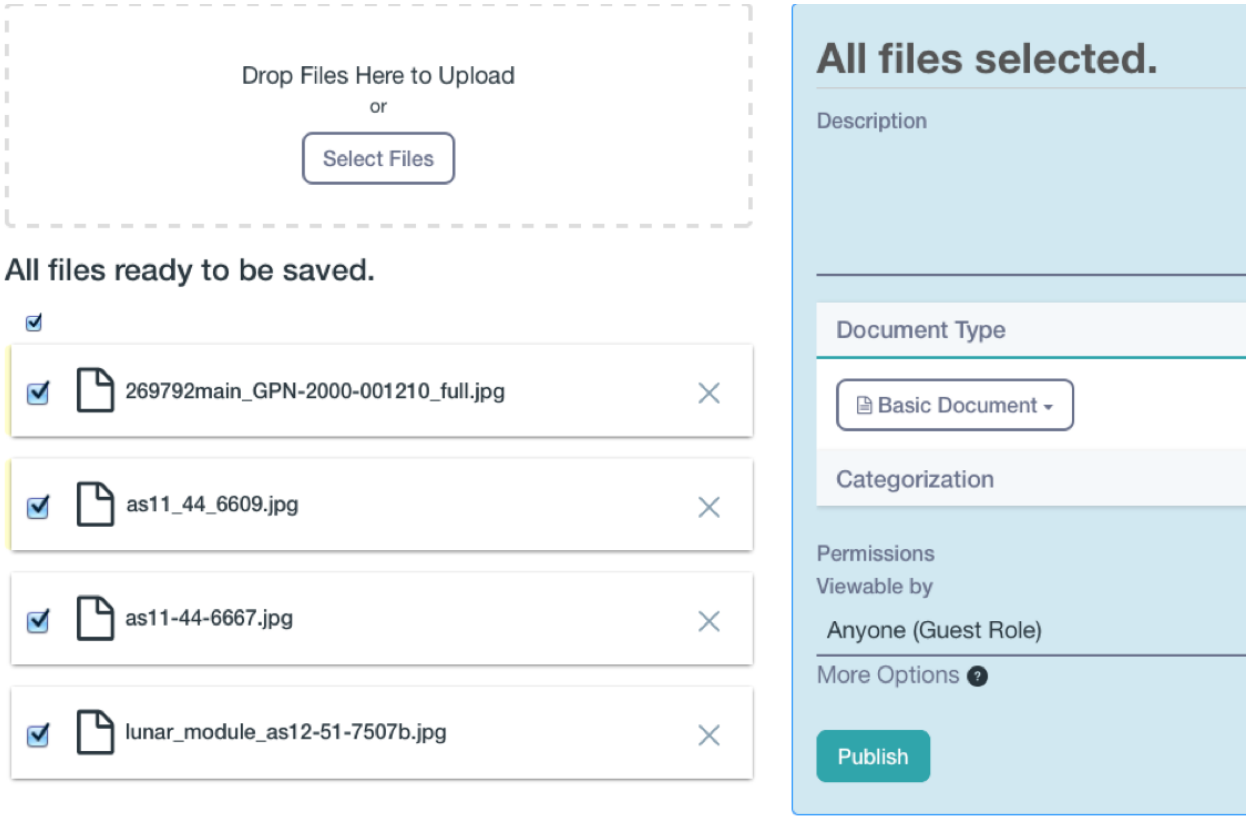


Figure 16.4: Documents and Media conveniently lets you add multiple files at once.

image. If the file is an image type file, the image is used as the file entry's cover image. If the file isn't an image, Documents and Media displays a generic image for that file type. Also displayed with each file icon is the file's suffix (e.g., *JPG* for a file ending in `.jpg`), last modification timestamp, name, and status (e.g., *Approved*, *Draft*, etc.). And each file has an Actions menu. The actions are also available from within the file entry's view, which we'll explore soon. The descriptive and list display styles provide the same functionality, but display the file entries in rows.

To see the current folder's details click the *Information* icon ( ). An area with the folder's name appears. It shows the number of items in the folder. It also shows a *Subscribe* icon ( ) you can select to get notifications about files added to or modified in the folder. Lastly, the Actions icon lists actions you can perform on the current folder. Above the Information icon and Display Styles is a *Search* that enables you to find files by keywords.

On the other side of the screen also in the top area is a filtering selector. It's default option is *All*. Click it to see the other filter options. The *All* option (default) shows all of the current folder's immediate subfolders and files. The *Recent* filter shows the most recently modified files, *Mine* lists all the current user's files (no matter their folder), and *Document Types* filters on files of the selected document type. If a Document Library contains more documents than it can display at once, you can use the navigation tool at the bottom of the app's window to either switch your view to another page or configure the page to display more documents per page.

Next to the filtering selector, the *Order by* selector lets you select criteria for ordering the files and folders. You can order them by creation date, title, number of downloads, modification date (default), or size. The up

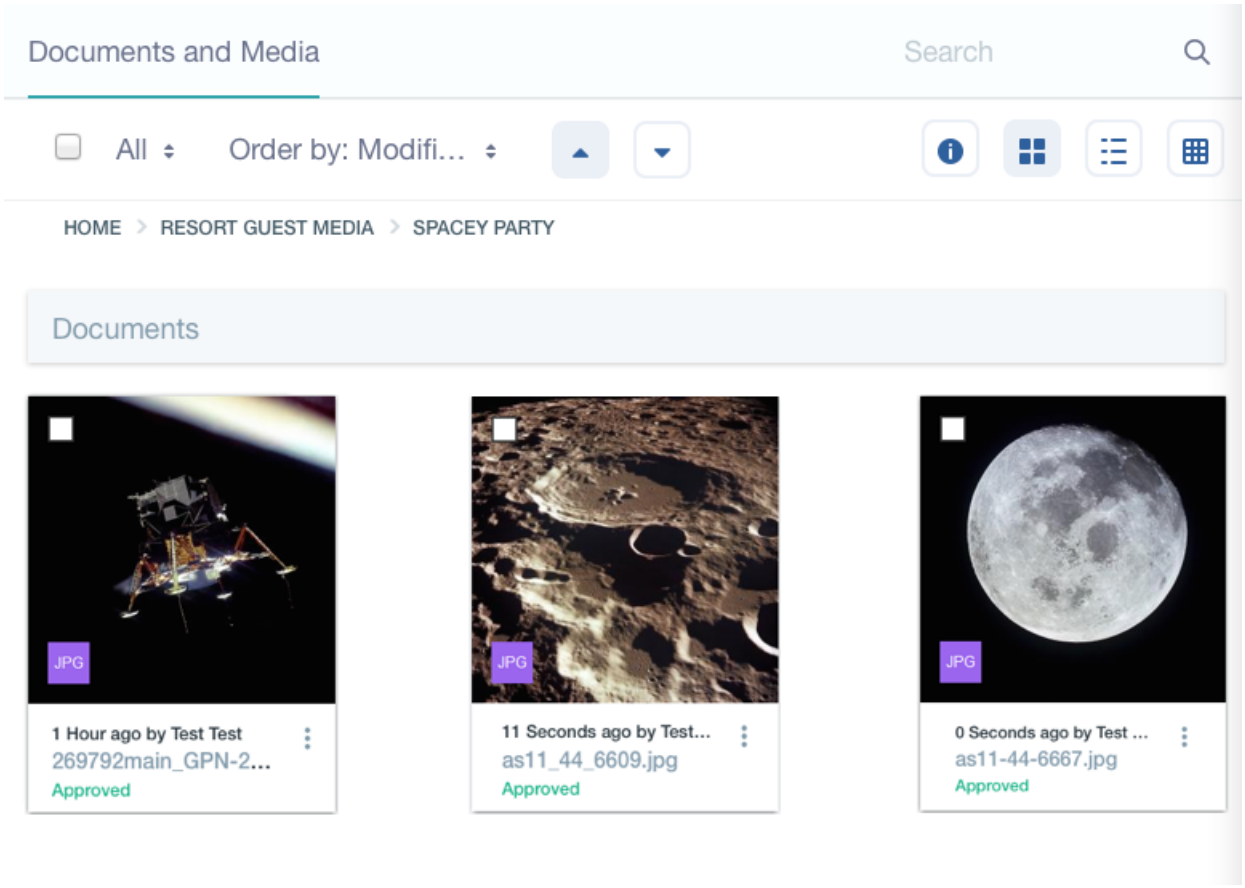


Figure 16.5: Documents and Media's lists files using several display styles: icon (as shown here), descriptive, and list. And breadcrumbs show the current folder's path in the library.

and down arrows next to the *Order by* selector, let you arrange them in ascending or descending order.

From the list view, Documents and Media lets you act on one or more files at once. When you select a checkbox for one or more files, the following action icons appear: . Describing them from left to right, the X is for canceling file checkout, the secured lock is for file check-out, the unsecured lock is for file check-in, the arrows are for moving the files, and the trash can is for moving files to the Recycle Bin. You can also move selected files to a subfolder via drag and drop. File check-out and check-in is explained in Collaborating on Files.

Viewing File Previews

You've added several files to the *Spacey Party* folder. In the Document Library, each file has a file entry view. To open its file entry view, click the file's name. The screen's central viewing area displays a preview image of the file. If the file is an image file, its image is displayed. If a preview plugin for the file type is installed, the plugin displays an image (e.g., the opening scene of a video file or a presentation's first slide) for the file. If there are no preview plugins for the file, Liferay DXP displays a generic image based on the file's type. Let's take a moment and consider file preview plugins and some of the powerful features they offer.

Whenever possible, Liferay DXP generates previews of documents added to the Document Library. Out of the box, Liferay DXP only ships with Java-based APIs to generate previews for documents. The only tool available that is 100% Java and has a compatible license to be distributed with Liferay DXP is PDFBox. If

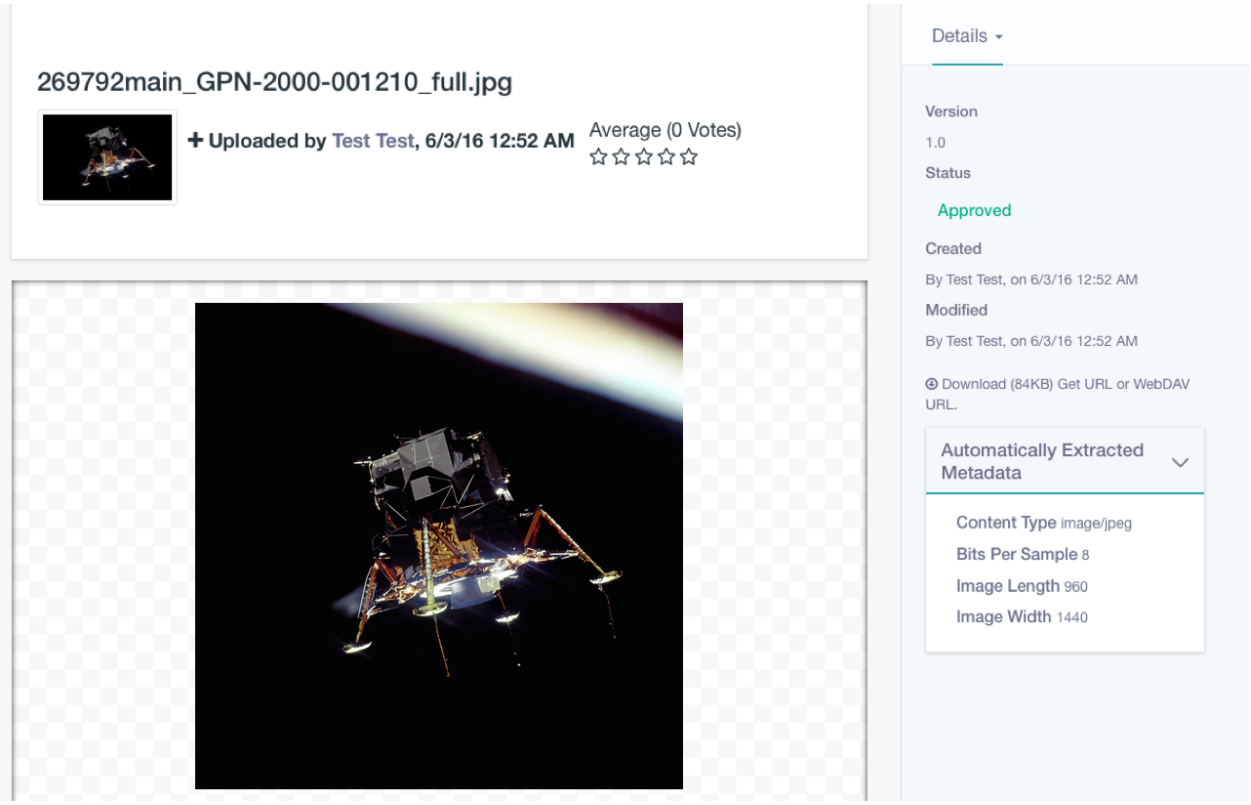


Figure 16.6: A file's entry view lets you act on the file, preview it, and inspect its details. If you've installed an appropriate preview plugin for a file, its preview image displays in the preview area. Liferay can, by default, preview many image types.

you upload a PDF file to Documents and Media, Liferay DXP generates a preview for the PDF in a separate thread. This process may last only a few seconds for a small file. The larger the file, the longer it takes.

While a default implementation of image generation for document previews and thumbnails is provided via PDFBox, you must install and configure some additional tools to harness the full power of Liferay DXP's Documents and Media library. These tools include *OpenOffice* or *LibreOffice*, *ImageMagick*, which requires *Ghostscript*, and *Xuggler*. With these tools installed and configured, Documents and Media content is displayed using a customized viewer depending on the type of content. Configuring Liferay DXP to use OpenOffice or LibreOffice in server mode allows you to generate thumbnails and previews for supported file types (.pdf, .docx, .odt, .ppt, .odp, etc.), lets you view documents in your browser, and lets you convert documents. ImageMagick allows for faster and higher-quality previews and conversions. Xuggler allows for audio and video previews, lets you play audio and video files in your browser, and extracts thumbnails from video files. You can configure the tools via portal properties you can set in the Control Panel's Server Administration screen or in a `portal-ext.properties` file. To learn how to use these tools, refer to Configuring Liferay DXP.

You can view a document with a customized viewer that allows you to navigate through the different pages of the document and read its content. In addition, you can view a multimedia document (audio or video) and play it online. If the browser supports HTML5, it uses the browser's native player. Otherwise it falls back to a Flash player.

Document previews are powerful and help users browse media more successfully to find what they're looking for.

Above the file viewing area are the file's icon and the file entry's name, author, upload timestamp, and

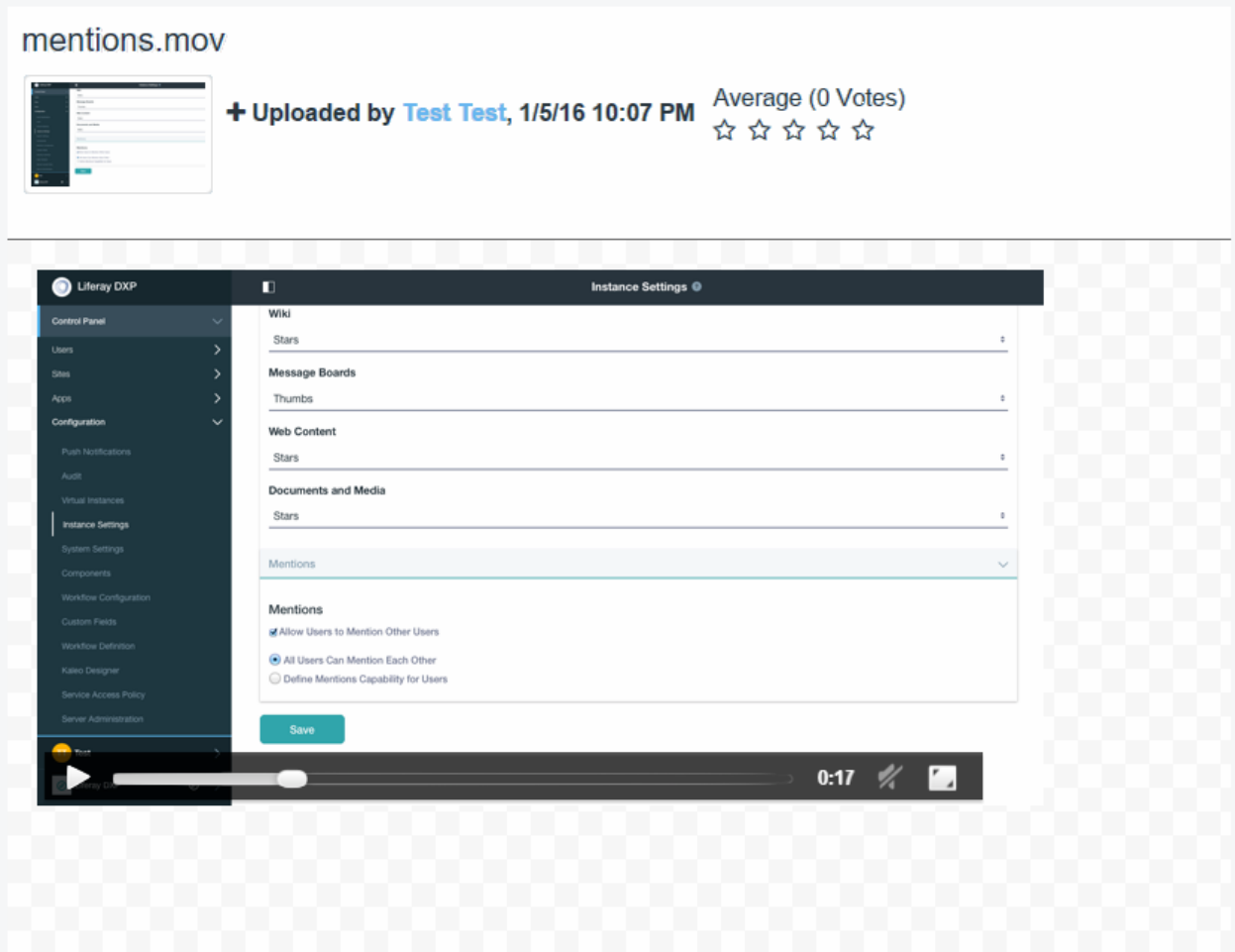


Figure 16.7: By leveraging services like Xuggler and ImageMagick, you can watch a video clip or even view a slideshow inside Liferay DXP's Documents and Media app.

rating. You can rate the file too. The comments area below the file viewing area lets you comment on the file, subscribe to comments, and reply to comments.

A file's options are accessible from the *Options* icon ( ) at the top of the screen. Here are the file options:

Download: Downloads the file to your device.

Edit: Lets you modify the file's name, description, document type, categorization, and related assets. You can even upload a new file to replace it. Note, modifying the file increments its version.

Edit With Image Editor: Opens the Image Editor to modify the image. The Image Editor is explained in Editing Images.

Move: Relocate the file to a different parent folder.

Checkout/Checkin: Checkout prevents others from modifying the document while you are working. Other users can still view the current version of the document if they have permission. You can check in the document when you're done working.

Permissions: Lets you specify which actions each role can perform on the file. You can granted a role permission to perform the following actions.

- **Update**: Edit, checkout, and checkin the file.

- **Override Checkout**: Checkout the file, revoking the file's current checkout if it is checked out.
- **Permissions**: View and configure this file's permissions.
- **Delete**: Move the file to the Recycle Bin.
- **View**: View the file.
- **Update Discussion**: Edit another user's comment on the file.
- **Delete Discussion**: Delete any comments on the file.
- **Add Discussion**: Comment on the file.

Move to the Recycle Bin: Remove the file from the Documents and Media library to the Recycle Bin.

Click the *Information* icon ( ) to view the file entry details. The top portion of this area lists the file's version number, status, modification timestamp, and name of the user who last modified it. Next are links to download the file and links to show its URL and WebDAV URL. You can specify the WebDAV URL as the file name when opening the document from a desktop environment.

The section *Automatically Extracted Metadata* lists any and all metadata that's been extracted automatically from the file. When adding new documents or viewing existing documents, a process is triggered automatically that extracts the file's metadata. The library used by this process is TIKA and it's already included in Liferay DXP out of the box. Depending on your file's type and the metadata written with the file, you can find out all kinds of details. In the case of audio or video files, their duration is displayed.

To view the *Version History* click the drown-down arrow next to *Details* and select *Versions*. This section lists the different versions of the file and lets you view, download, remove, and revert to specific file versions. File version history actions are explained in the Collaborating on Files guide.

Let's review what you've done so far. First, you created a role to manage a specific set of files for your site. You assigned users to the role and created a Documents and Media folder named *Resort Guest Media* for them to add and organize files. Then as a member of the role, you added a subfolder named *Spacey Party* and added files to it. And just now, you viewed individual file entry information and actions.

Next, you'll learn how to edit images in Liferay DXP.

16.2 Editing Images

You've uploaded an image to the document library. It's almost perfect, but it's missing a little something. It could use a little more focus to really highlight the message you're trying to communicate. Just a little editing and the image would draw your readers in. You could edit the image in your favorite image editing program and re-upload it, but that's a bit tedious for the minor changes this image requires.

What else can you do?

Lucky for you, Liferay DXP now comes with its own built-in image editor.

A crop here and there, a little bit more contrast, and some adjustments in saturation, and boom! The image is ready to go.

Since Liferay Portal CE 7.0 GA2 and Liferay DXP 7.0, users can edit images within portal instances. To access the image editor, go to the Documents and Media repository and locate the image you want to edit. Click the Actions icon ( ) and select *Edit With Image Editor*.

The image editor can also be accessed through the item selector. Essentially, anywhere you have an image, you can edit images. For example, in blog entries, web content articles, and wikis, you can view a preview of the image.

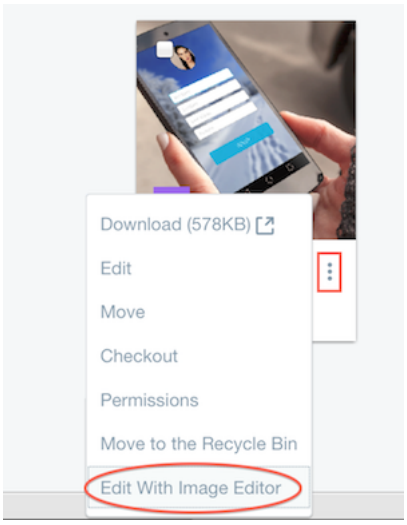


Figure 16.8: You can access the image editor through the Documents and Media repository.

Within the item selector preview window, you can access the image editor by clicking the pencil icon ( ) in the bottom right corner of the preview window.

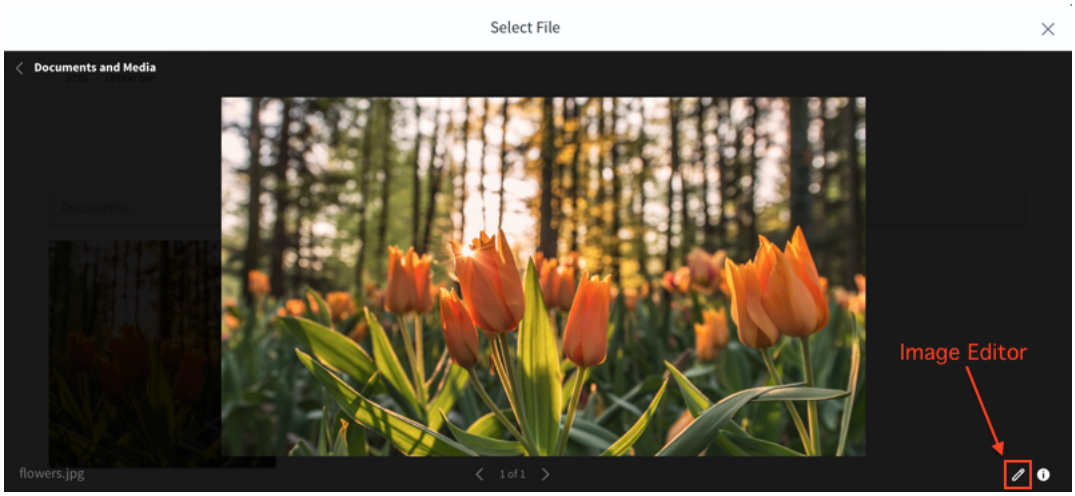


Figure 16.9: You can access the image editor through the item selector preview window.

The image editor was designed with quick editing in mind, and offers a minimal, user-friendly UI:

Figure 16.10: The image editor's UI is clear and to the point, offering only what you need.

The main toolbar consists of three buttons, each of which contain a subset of options. The menu options are described below:

Transform Tools ( )

Crop: Lets you reframe the image, by removing the outer portions.

Resize: Lets you resize the image in pixels. If the lock is closed, the aspect ratio remains locked, and changing either width or height will automatically adjust the other dimension to maintain the aspect ratio. When the lock is opened, the width and height can be changed individually, allowing the aspect ratio to be altered. This is not recommended though, as the image can become distorted.

Rotate: Lets you rotate the image to the left or right, in 90 degree increments.

Adjustment Tools ( )

Saturation: Lets you adjust the amount of color saturation in the image. The default value of 50 is the current saturation. The slider values range from 0(completely desaturated) to 100(completely saturated).

Contrast: Lets you adjust the difference between highlights and shadows. A value of 50 is the current level of contrast. The slider values range from 0(no contrast) to 100(full contrast).

Brightness: Lets you adjust the amount of light in the image. A value of 50 is the current brightness. The slider ranges from 0(completely black) to 100(completely white).

The third and final toolset is a preset of filters, which you can access by clicking the wand icon ( ):

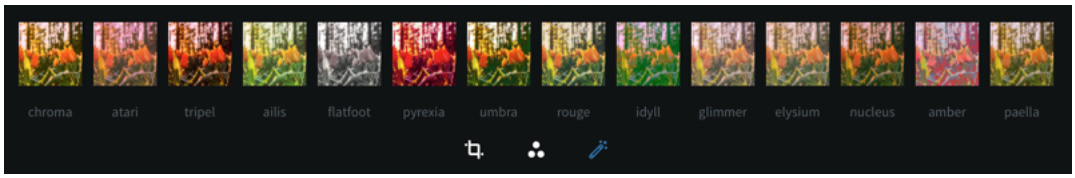


Figure 16.11: The image editor offers several filter presets for you to choose from.

Once you have made changes to the image in the editor, you can click the *Cancel* button to cancel the changes, or the *Apply* button to apply the changes to the image. Once you have applied changes to your image, the history bar appears:

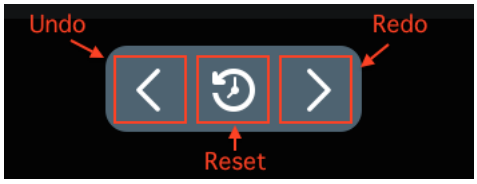


Figure 16.12: The history bar allows you to undo, redo, and reset changes.

The history bar allows you to undo, redo, and reset changes that have been made to the image in the editor. Note that the reset button should be used with caution; once it has been pressed the image is reset to its original state, causing all changes made within the editor to be lost.

As mentioned earlier, you can access the image editor in the Documents and Media repository, or through the item selector in blogs, web content articles, and wikis. When the image is edited within the Documents and Media repository, and saved, the version is incremented a minor version i.e. version 1.0 to version 1.1. You can view the previous versions of the image through the *version history* table, under the information section for the image. When the image is edited within the item selector, and saved, a copy is created of the image and saved to the document library.

Next, you'll learn how to access Google Docs™ through the Document Library.

16.3 Accessing Google Docs™

You can access files stored in Google applications, such as Google Drive™ and Google Photos™, through the Document Library. This integration isn't available by default, but you can get it by installing the Liferay

Plugin for Google Drive™ from the Liferay Marketplace.

Important: The Liferay Plugin for Google Drive™ is a Labs application available for Liferay DXP 7.0 and Liferay Portal CE 7.0 GA6+. Apps designated as Labs are experimental and not supported by Liferay. They're released to accelerate the availability of useful and cutting-edge features. This status may change without notice. Please download and use Labs apps at your own discretion.

You must configure Liferay DXP to access Google's Picker API, which accesses files stored on Google's servers. Invoking the Google Picker API requires unique API keys and a client ID. The API keys are for accessing public data and the client ID is for authenticating and authorizing user access to non-public data. To get the keys and client ID, you must use a Google developer project. This takes three steps:

- **Step 1: Enable the Google Picker API**
- **Step 2: Create Credentials**
- **Step 3: Configure Liferay DXP's Google Apps Settings**

Start by enabling the Google Picker API in a Google developer project.

Step 1: Enable the Google Picker API

Liferay DXP uses the *Google Picker API* to access files in Google.

1. Create a Google account or use an existing one.
2. Open the Google Developer Console.
3. Create a new project or use an existing one.
4. Search for the *Google Picker API* and enable it. Liferay DXP uses this API to access Google documents.

Next, you must create credentials to use with the Google API.

Step 2: Create Credentials

1. In the Google Developer Console, navigate to the *Credentials* screen for APIs & services.
2. Create a new OAuth client ID for a *Web application*. Specify the following attributes:
    - **Name**: `Google Docs Hook`
    - **Authorized JavaScript origins**: `[liferay-instance-URL]` (e.g., `http://localhost:8080` by default for local development machines)
    - **Authorized redirect URIs**: `[liferay-instance-URL]/oath2callback`
3. Create a new API key. Restrict the key to HTTP referrers (web sites), and set it to accept requests from your Liferay DXP instance's URL.

Your new OAuth client ID and public API access key now appear on the Credentials screen. Keep this screen open to reference these values as you specify them in Liferay DXP. All that's left is to enter the API key and client ID into Liferay DXP's Google Apps configuration.

Step 3: Configure Liferay DXP's Google Apps Settings

To call Google's APIs, you must configure Liferay DXP with your Google project's OAuth client ID and public API access key:

1. Navigate to *Control Panel* → *Configuration* → *Instance Settings*.
2. Click the *Miscellaneous* tab and expand the *Google Apps* section.
3. For *Google Apps API Key*, enter the Google API key that you created in the previous step.
4. For *Google Client ID*, enter the Google OAuth client ID that you created in the previous step.
5. *Save* your changes.

Your Liferay DXP instance is now ready to access Google documents from the Document Library.

Linking to Google Docs™

In your Document Library, you can create file entries that link to Google documents, such as files stored in Google Drive™ or photos saved to Google. To link to a Google document in Documents and Media, click the Add icon ( ) and select *Google Docs*. The *New Google Docs* screen appears.

This screen is similar to the edit screen for basic documents, except for file selection. Clicking the *Select File* button brings up Google's file picker to select files from your Google Drive™ or your photos. Select the Google file you want to link to and click *Publish*.

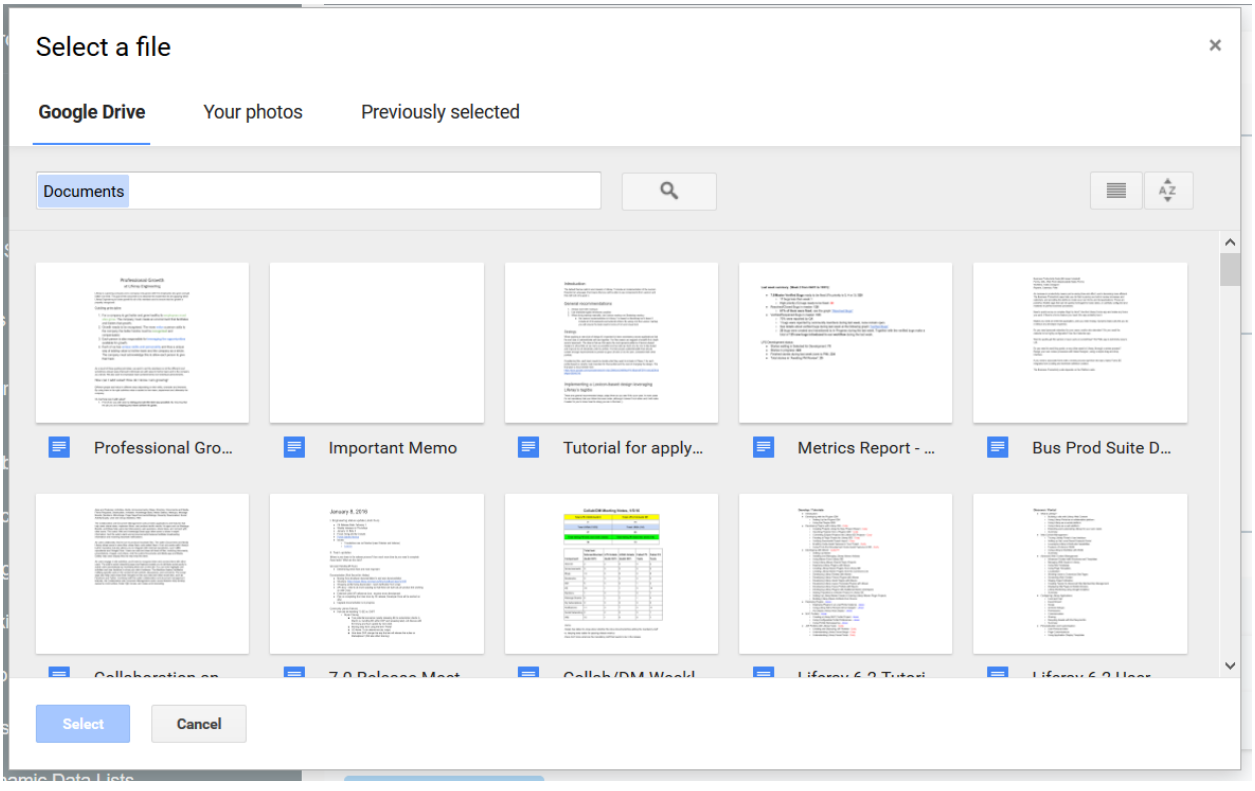


Figure 16.13: You can select files from Google Drive™ or your photos.

A new file entry appears for the Google document you linked to. You can view the file entry as you would any file entry. The Google document's contents show in the file entry's preview pane. As with any file entry, the *Options* button ( ) gives you access to the Download, Edit, Move, Permissions, Move to Recycle Bin, and Checkin/Checkout/Cancel Checkout options.

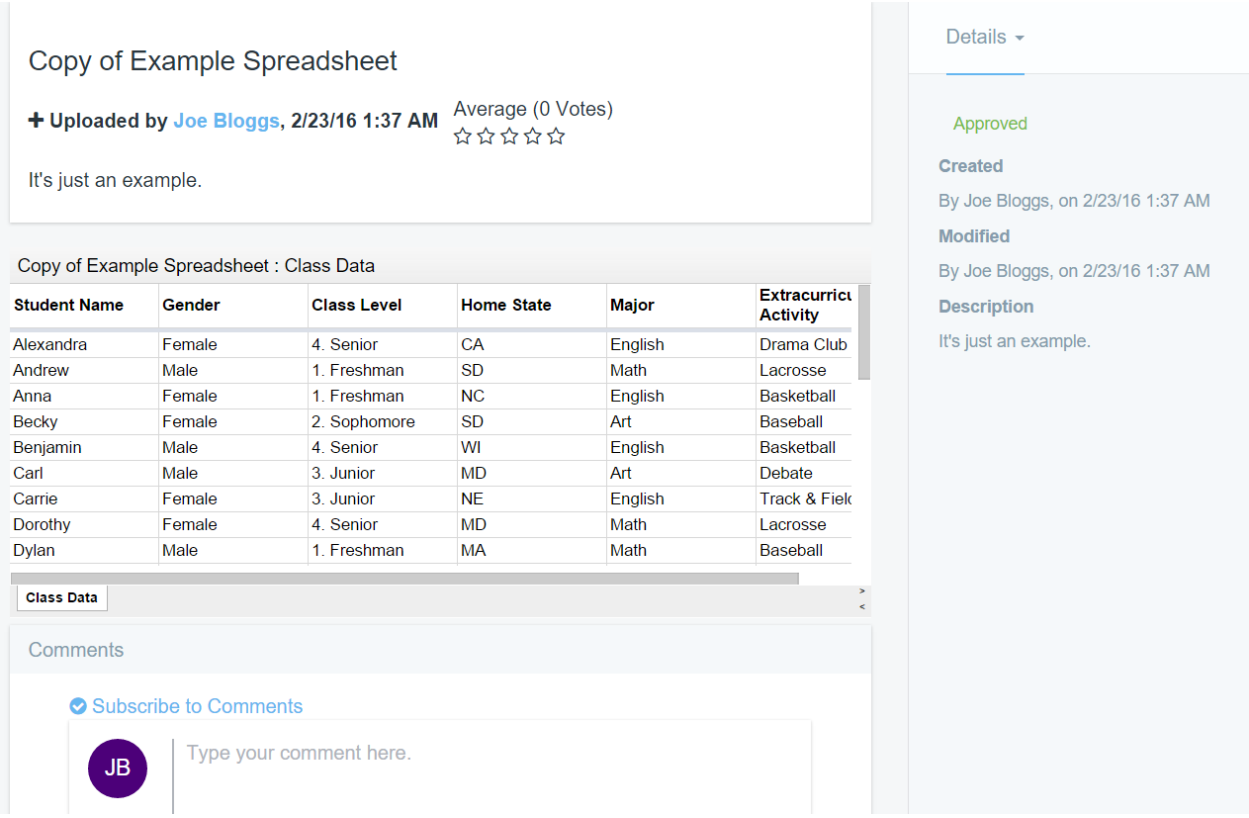


Figure 16.14: The Google document's file entry view displays the file's information and provides a preview of the file.

Great! Now you know how to access Google Docs™ from your document libraries. Next, you'll learn how to display your Document Library's files on your site.

16.4 Displaying Files

Once you've uploaded image files and videos to your Document Library, you'll probably want to display them in your site. Here are some ways you can display your media files:

- Show them in a Documents and Media app
- Display them in a Media Gallery
- Use the Asset Publisher
- Insert them in an asset, like a Web Content article or Blogs Entry

Continuing with the Lunar Resort example, the media team sits down with each guest party to show them a page filled with photos and videos of their vacation experience. The team likes to show guests a slideshow to get some oohs and ahh's from them and determine their favorites. Then they offer to print and frame the photos and plaster the photos on all kinds of fun items such as keychains, coffee mugs, and

tee-shirts. They even offer to burn media files onto DVDs. Let's explore how the media team shows off the guest's pics and videos on a site.

Creating a Media Page

The media team creates a page for each guest party. Team members can customize each page to focus on each party's media. Each party's media page will have a Media Gallery app. The Media Gallery shows a large thumbnail of each media file, lets the user download individual files, and comes with slideshow capabilities. Let's create a media page for the Spacey guest party and add the Media Gallery.

Add a page named *Spacey Party*. Next, click the *Add* icon ( ), then navigate to *Applications* → *Content Management*, and select *Add* next to *Media Gallery*. The app appears on the page and shows your Home folder's files by default. Since we want to focus on the Spacey party's files, let's configure the app to show files from the *Spacey Party* folder. Click the app's Options icon ( ) and select *Configuration*. The Configuration window appears and shows the *Setup* tab.

The Setup tab's Display Settings section provides checkboxes to enable showing each file's actions, gallery navigation links (for paging through media files), and a search field. File search is enabled by default. The Media Type selector lets you filter on media file types; all supported types are filtered on by default. And the Media Gallery uses a Display Template. You can create your own Display Templates for the Media Gallery or select an existing display template, like the *Carousel* display template.

Important: To access the Carousel display template in Media Gallery, your role must have *View* access for that template. Since the Carousel template is in the Global scope, a Global scope administrator must grant the role permission to view the template.

The Folders Listing section lets you select a Document Library folder to serve as the root folder from which to display files. The root folder is the highest-level folder that's accessible from the Documents and Media app. For example, suppose you created a folder called *My Documents* in Documents and Media's default Home folder. If you set the My Documents folder to be your app's new root folder, the app accesses the My Documents folder and no longer accesses the Home folder.

Lastly, the Ratings section of the Display Settings lets you enable users to rate files and comment on them.

Here are descriptions for the Media Gallery app's other configuration tabs:

Communication: lists public render parameters the application publishes to other applications on the page. Other applications can read these and take actions based on them. For each shared parameter, you can specify whether to allow communication using the parameter and select which incoming parameter can populate it.

Sharing: lets you embed the application instance as a widget on on any website, Facebook, or Netvibes, or as an OpenSocial Gadget.

Scope: lets you specify the Document Library instance the application uses: the current site's instance (default), global instance, or the page's instance. If the page doesn't already have an instance of the app, you can select scope option *+ [Page Name] (Create New)* to create a page-scoped instance for the application to display.

While we're in the Configuration window there are a couple things we should do. To enable the media team role member to act on the files, select the *Show Actions* checkbox. Since we only want to show the Spacey's files, expand the *Folders Listing* section, then click the *Select* button to browse and select the *Spacey Party* folder. Then click *Save*.

The Media Gallery now shows all of the Spacey party's files. This app enables the Lunar Resort media team to act on individual files. If Mrs. Spacey likes particular files, the staff member can download them locally and add notes in each file entry's comments.

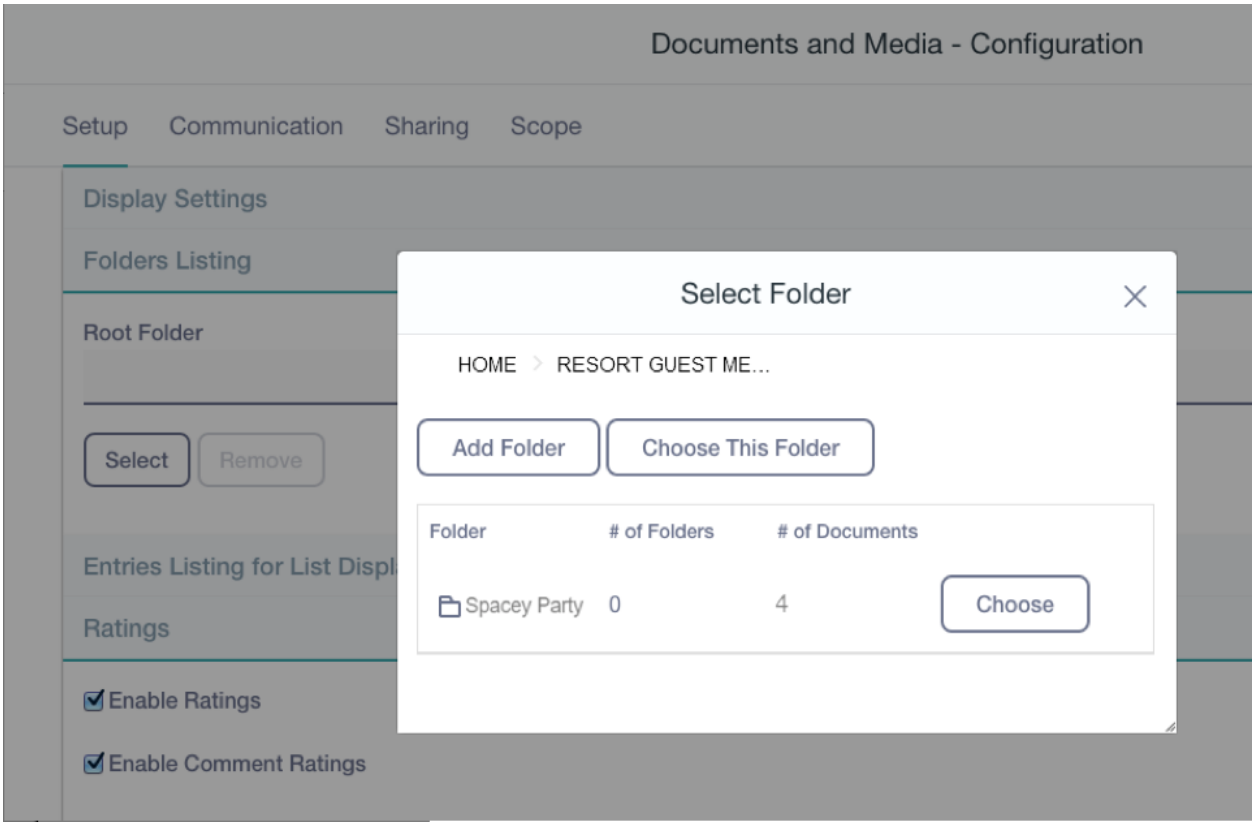


Figure 16.15: The Documents and Media app can be configured to use any folder as a root folder to display.

The Spacey Party's large image thumbnails display in the Media Gallery. When you click on a file's thumbnail, a slideshow appears showing the selected image. The window lists thumbnails of the folder's other images below the image that's currently shown. And the slideshow continues until you either click pause or view the folder's last image. The slideshow feature provides a nice way to show off images. Click the X to close the slideshow window and return to the page.

The Media Gallery and Documents and Media apps are at your service to show off your Document Library files. The Lunar Resort's Souvenir and Memorabilia group enjoys patron reactions to the great pics and videos that Snappy takes of them. And the salespeople are more than happy to plaster the pictures onto clothing and accessories to make a handsome commission.

Next, you'll learn how to collaborate on files with other users.

16.5 Collaborating on Files

Not only does Liferay DXP's Documents and Media enable you to publish files easily, but it also provides a terrific environment for collaborating on documents. There are many common collaboration scenarios. For example, you might have a spreadsheet that you and your peers need to populate, or a proposal that you and other stakeholders must produce. You might be asked to review and edit a document. All these cases involve multiple users modifying documents and require an environment that helps manage document changes. Liferay DXP's Documents and Media provides these capabilities.

Liferay DXP's document management apps support accessing apps on different kinds of devices and

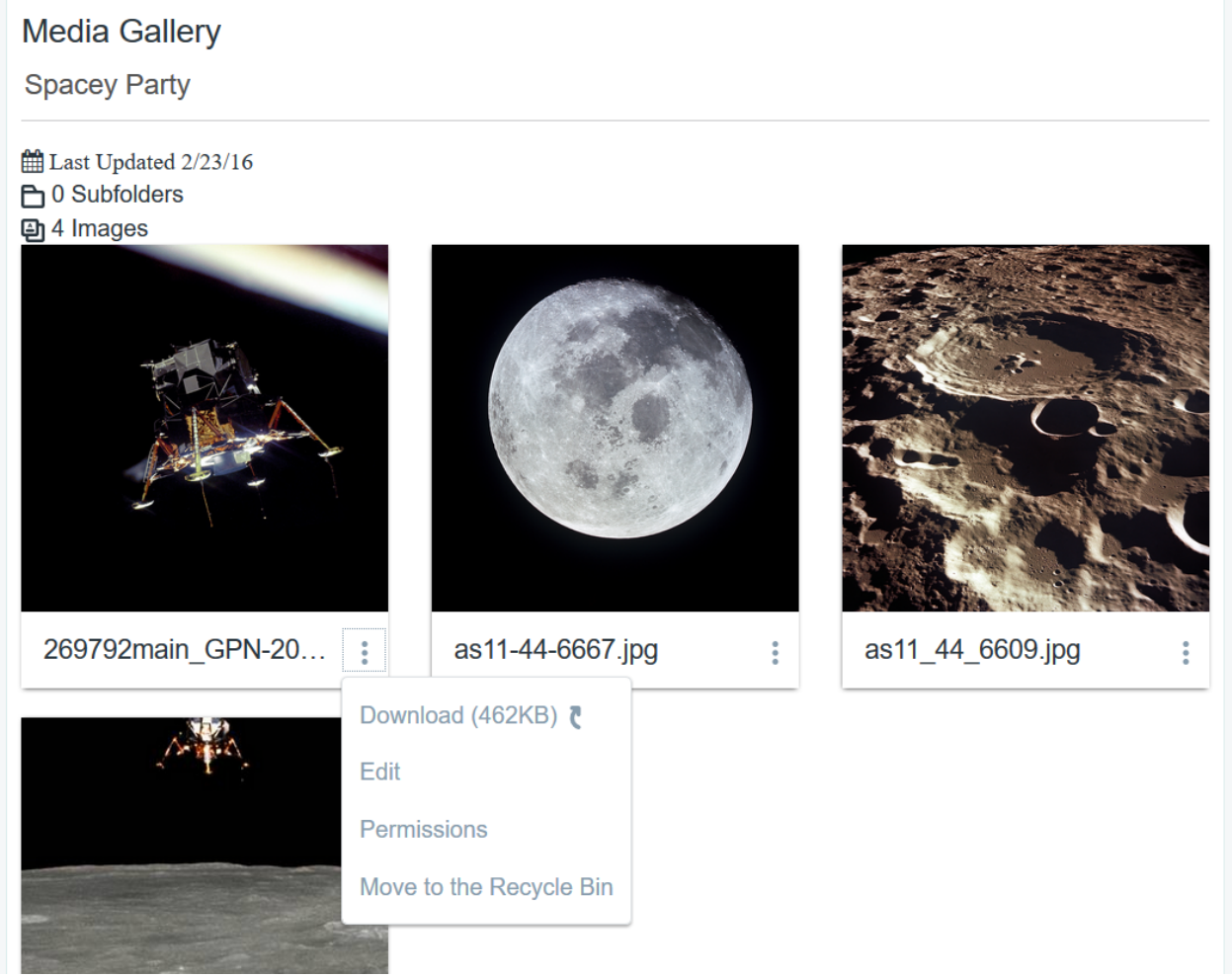


Figure 16.16: The Media Gallery renders large thumbnail images of media files. It gives users quick access to download files that interest them.

environments. Not only can you access documents from your site via your browser, but you can also access them from your native file browser and use Liferay Sync to access them from your tablet or mobile device. And Liferay Sync makes sure you're you can access the latest document versions.

To demonstrate using these features, we'll join the Lunar Resort Recreation Department members collaboration on an important document: the Lunar Resort's upcoming activities newsletter. Resort representatives from each activity group must contribute content to entice resort guests to sign up for activities. Staff members involved include Chip Shot (golf instructor), Wilma Carwin (lunar rover race instructor), and Tenzing Nepal (lunar sherpa). Each of them use different devices and bring their own style and attitude to the collaboration. So perhaps you'll be able to relate to their effort.

Before the group leads can start working on the newsletter, the department director (who's associated with the organization's content administrator Role) creates a folder called *Newsletters* in the organization's document library. To follow along with the example, create the *Newsletters* folder in an organization's document library on your site. This folder will hold all of the department's newsletters.

Next, open your favorite document editor, enter the title "Upcoming Lunar Activities", and save the document as `upcoming-activities.docx`. Upload the file to the document library and enable the leads to edit the file by associating them with Role that you grant permission to edit the file. To enable the activity leads

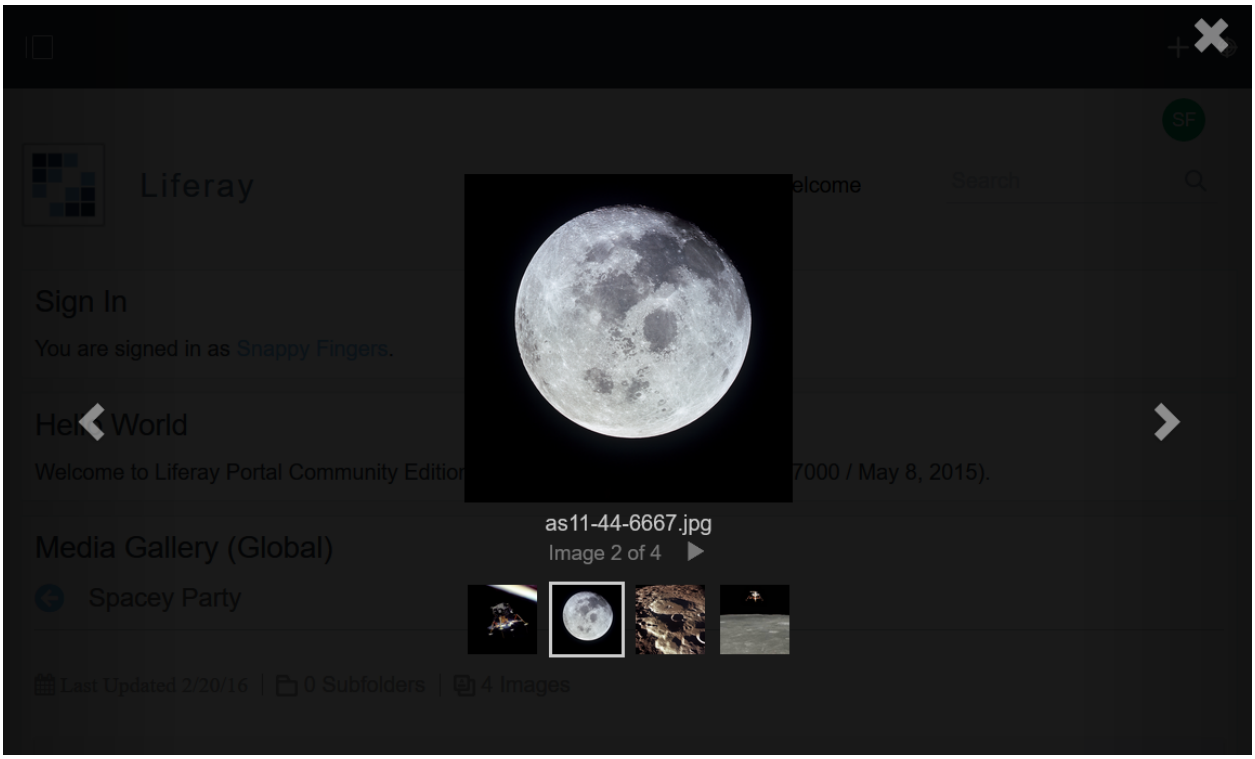


Figure 16.17: The Media Gallery's slideshow provides a nice way to view images.

to access the newsletter from the department's private Organization Site, add a Documents and Media app to its home page. Then turn on the *Show actions* display option in the app's configuration window. Make sure to assign the leads a role with permissions to access the folder. From the Documents and Media app on Recreation Department's private home page, the leads can now access the *Newsletters* folder and the initial document. The newsletter's next collaborator will be Tenzing Nepal, the resort's veteran lunar sherpa.

Tenzing is as agile online as he is on a mountainside. He often edits his documents across several versions so they read just as he likes them. He prefers to check documents out from within the web interface and download them to his computer to work on them. Let's follow Tenzing as he produces multiple versions of the newsletter.

Creating New File Versions

To prevent other users from accidentally overwriting changes you're making to a document, the Document Library allows you to *checkout* the document. The user who checks out a file is the only user who can checkin changes to the file. Checking in a file produces a new file version.

Tenzing Nepal is the first resort instructor to edit the activities newsletter. You can imitate Tenzing's actions or follow along as you create new versions of your own files. Inside the Documents and Media app, select the `upcoming-activities.docx` file's *Checkout* action. Upon checkout, the file's status is *Draft*. While the file is checked out, you're the sole editor; all other users must wait until after you've checked in the file to edit it. While you have the file checked out, you can download it, replace the file entry's file, move the file entry to another Document Library folder, check it in, or cancel checkout. Tenzing proceeds with the next step towards contributing to the file.

Unless you're using Liferay Sync or a local drive mapped to the file's WebDAV (explained in the next section) URL, you'll typically follow the basic workflow below to edit a Document Library file from your machine:

1. Checkout the file.
2. Download the file and edit it locally.
3. Edit the file locally.
4. From the file's Edit screen in Documents and Media, browse to and select the local file. Save and Checkin
5. Check in the file.

Let's continue as Tenzing, following the above worklow. (1) You've already checked out the file. (2) Download the file to your machine by clicking the *Download* action. (3) Then open the file in your favorite document editor, add information on the upcoming Lunar hikes, and saves your changes. You're now ready to apply the changes to file in Documents and Media.

In the file entry's screen in Documents and Media, click on the *Edit* action. Then click *Browse*, navigate to the copy of the `upcoming-activities.docx` file you edited on your desktop, and select it. Then click the *Save and Check In* button and the *Describe Your Changes* screen appears. This screen allows you to describe your changes and specify whether to apply a major or minor version increment (default). For example, if a file's current version is 1.0, the file's next minor version is 1.1 and next major version would be 2.0. To increment the file to a new *Major Version*, you must select the *Changes Create a Major Version* checkbox. Your changes will otherwise result in a new Minor Version.

Select the *Changes Create a Major Version* checkbox, describe your changes in the *Change Log* field, and click *Save*. The file's major version number is incremented and its status is now *Approved*.

Note: If you *Edit* a file without checking it out, the Edit screen displays a checkbox *Customize the Version Number Increment and Describe My Changes*. For your file modification, if you want to select between major/minor version types and/or want to enter a change log then select the checkbox. On clicking *Publish*, the *Describe Your Changes* screen appears so you can specify your change's version increment type and description.

Since the file is checked in, any user can now check it out. Tenzing realizes there are some minor details he wants to add. As Tenzing, checkout the file once again. Download the file locally, make the changes (any changes you like), and save the local file. Then edit the file entry in Documents in Media, browse to and select your local file, and click *Save and Check In*. This time, leave the major version selection unchecked and click *Save*. The file is checked in and it's minor version number is incremented.

In the file entry's screen, the *Version History* table lists the versions of the file. The department director created version 1.0, Tenzing created major version 2.0, and Tenzing also created minor version 2.1. Each file version has an Actions icon (⋮). You can perform the following actions on each file version that's been checked in and approved (if no workflow is associated with a folder, the file's approved on checkin):

Download: Download the version of the file to your machine.

View: View the file entry screen for the version of the file.

Revert: Check in a new major version of the file based on that file version.

Delete Version: Remove the file version from the Document Library. All other file versions remain intact.

You've used the basic Document Library workflow to edit files locally and upload them to create new file versions of the newsletter. Next up is accessing documents from your desktop file manager. To demonstrate this, you'll step into the shoes of lunar rover race instructor Wilma Carwin.

Wilma updates the newsletter in a different manner. She doesn't care much for high tech. The only online device she uses is the workstation in the garage office. Wilma only uses it when she has to; otherwise

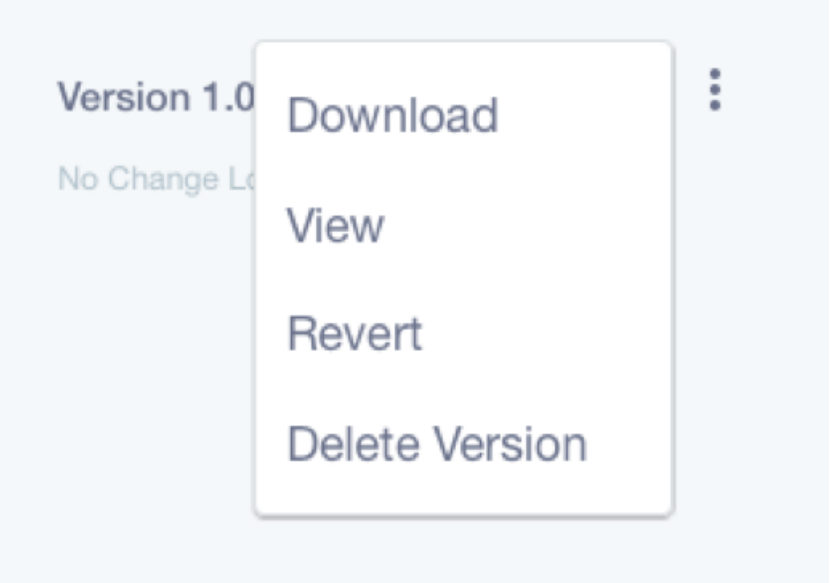


Figure 16.18: No file is "written in stone". Version history actions let you inspect, delete, and reinstate file versions.

she'd rather be behind the wheel of a rover doing donuts in the dirt and banking turns off the canyon walls. Fortunately for Wilma, the resort IT team mapped the rec department's document library to a network drive on her workstation. This lets her edit documents without having to meddle with the Lunar Resort site. It's worth explaining how they integrated her file manager with the document library.

Desktop Access to Documents and Media

You can access the Document Library from your computer's file manager via WebDAV. WebDAV stands for Web-based Distributed Authoring and Versioning. It's a set of methods based on HTTP that allows users to create, edit, move, or delete files stored on web servers. WebDAV is supported by most major operating systems and desktop environments, including Linux (both KDE and GNOME), Mac OS, and Windows.

At the Lunar Resort, Activity Department members use WebDAV to access their Document Library. WebDAV provides an alternative way to do this using a file manager instead of a web browser. To access the Document Library folder from your file browser, you need to use log in credentials for the Liferay DXP instance and the WebDAV URL of the folder you'd like to access.

Navigate to the Documents and Media app hosting the folder you'd like to access. Click on the folder's Actions icon ⋮ and select *Access from Desktop*.

Copy the WebDAV URL. On Windows, map a network drive drive to the WebDAV URL. Enter your Liferay DXP credentials when you're prompted for them. The document library folder appears in the network drive. From your file browser, you can now add, edit, move, or delete files in this folder.

On Mac OS X, select *Go → Connect to Server* in Finder. Then enter the WebDAV URL of the folder you'd like to access in the Server Address field, click *Connect* and you should be prompted for your Liferay DXP credentials.

On Linux, you must slightly modify the WebDAV URL of your folder in your file manager. For KDE's Dolphin, change the URL's protocol so that it says `webdav://` instead of `http://`. For GNOME's Nautilus, change the URL's protocol so that it says `dav://` instead of `http://`. Then press *Enter* and you're prompted for your Liferay DXP credentials.

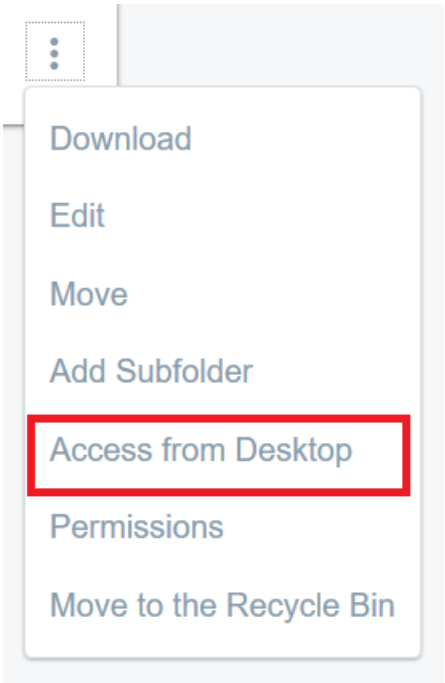


Figure 16.19: Select *Access from Desktop* to get the folder's WebDAV URL.

Note that Liferay DXP increments the version numbers of files edited and uploaded via WebDAV so you don't have to worry that using your file manager will bypass the functionality of Liferay DXP's web interface.

Lunar rover race instructor Wilma Carwin doesn't mind editing her Documents and Media files from her computer's file manager, because it's convenient and easiest for her. She opens the document `upcoming-activites.docx` and adds news about various types of rover racing lessons offered and the upcoming Lunar Grand Prix. She saves the file and closes it. The file in Documents and Media is now "Approved" and its minor version is incremented. Voila! She's done her part collaborating on the newsletter.

Following Wilma Carwin's demonstration, you've learned how to map your local file manager to a Document Library folder and collaborate on files from your desktop environment.

16.6 Publishing Files Summary

You've learned the fundamentals of uploading, managing, and collaborating on files in your sites. The Document Library integrates each file into Liferay DXP, allowing you to apply permissions, workflows, and rich metadata. And it provides a version control system for managing document progress. You displayed image files on pages using the Media Gallery and Documents and Media apps. And you've mastered checking out and checking in new file versions using the basic download/upload technique and by mapping your local file manager to the Document Library, for seamless file editing. Congratulations on your document management accomplishments!

16.7 Using Document Types and Metadata Sets

If you're concerned only with a file's content and are happy with its basic built-in metadata, then add the file to Documents and Media as a Basic Document. What do you do, however, if you want to ascribe more characteristics to a file? For example, you might have documents to which you want users to apply copyrights

and license information. Or you might want to add searchable attributes to documents. Is there a way to do these things in Documents and Media? Yes! Liferay DXP enables you to define metadata for describing files. Liferay provides tools to create and aggregate metadata into Metadata Sets that you can apply to Document Library files. You leverage the Metadata Sets through Document Types. These Document Types wrap the Documents and Media files so users can completely describe them.

Documents and Media uses a Document Type to classify every Document Library file. A Document Type wraps each file with a type name and metadata. The metadata fields let authors describe the document. As a Documents and Media administrator you create metadata fields, add them to Metadata Sets, and apply the sets to various Document Types.

Typically you'll want to specify document metadata fields in a Metadata Set and then apply it to a Document Type. So let's start creating a metadata set.

Creating Metadata Sets

Metadata Sets aggregate metadata fields so you can apply them to Document Types. Liferay provides Metadata fields to capture textual, boolean, date, numerical, location, and descriptive characteristics about documents. When users create or edit a document of a document type that has a metadata set, they specify metadata field values to describe the document.

Liferay DXP supports the following metadata fields:

Boolean: is a check box.

Date: lets you enter a date. A valid date format is required for the date field, but you don't have to enter a date manually. When you select the date field a mini-calendar pops up which you can use to select a date.

Decimal: lets you enter a decimal number. The value is persisted as a double.

Documents and Media: lets you select a file from one of the Liferay DXP instance's Documents and Media libraries.

Geolocation: lets you specify a location to associate with the document.

HTML: an area that uses a WYSIWYG editor to enhance the content.

Integer: lets you enter an integer. The value will be persisted as an int.

Link to Page: lets you link to another page in the same site.

Number: lets you enter a decimal number or an integer. The value is persisted either as a double or an int, depending on the type of input.

Radio: displays several clickable options. The default number of options is three but this is customizable. Only one option can be selected at a time.

Select: is just like the radio field except that the options are hidden and have to be accessed from a drop-down menu.

Text: lets you enter a single line of text.

Text Box: is just like the text field except you can enter multiple lines of text or separate paragraphs.

The easiest way to create metadata fields is by creating them in Metadata Sets. In Documents and Media in Site Administration, select *Options* ( ) → *Metadata Sets* to open the Metadata Sets window.

You can view, edit, or add metadata sets from the Metadata Sets window. If you select the check box next to the *Order by* heading all the sets are selected. If you select one or more sets, the Recycle Bin icon ( ) appears. To move selected metadata sets to the Recycle Bin, click the Recycle Bin icon. The *Order by* selector lets you order the sets by *Modified Date* or *ID*. The up and down arrows let you arrange the sets in ascending or descending order, respectively. You can also click on a metadata set's name to edit it.

The following actions can be performed on each metadata set.

Edit: opens an editor for modifying the metadata set.

Permissions: grant or revoke permission for roles to update, delete, view, or configure permissions for the metadata set.

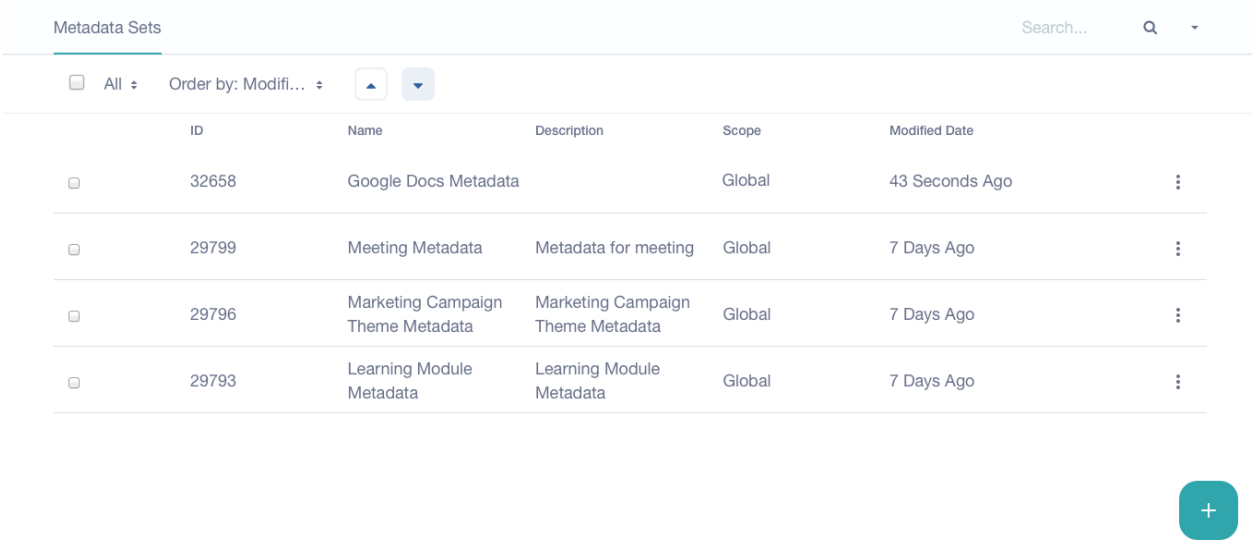


Figure 16.20: The Metadata Sets management window lets you view existing sets and create new ones for applying to document types.

Copy: Make a copy of the metadata set.

Delete: Move the set from Documents and Media to the Recycle Bin.

You create metadata sets independently of any document types, but you can apply them to any number of document types.

As an example, let's create a metadata set called *Syllabus Fields* for a document type called *Syllabus*. What metadata should we associate with syllabi? Let's choose for our syllabi to have course title, professor, location, semester, course description, and course requirements fields. All syllabi in our Liferay instance should be created using the *Syllabus* document type and authors must specify values for these fields. This ensures that a syllabus shows up in the Search app when keywords match its metadata field values.

To start creating the Metadata Set, click the *Add* icon ( ). The *New Metadata Set* window appears. Name the metadata set *Syllabus Fields*.

Note that you can translate each of a metadata set's field values to any of the locales your Liferay instance supports. To specify a field value for a translation, select the flag that represents the locale and enter the field value for the locale.

In the *Details* dropdown section you can describe the metadata set and select an existing metadata set to extend. If there's a metadata set to which you want to simply append more fields, then click the *Select* button and choose the metadata set you want as your new metadata set's parent. When a user creates adds a document of a document type that uses this metadata set, the parent metadata fields will appear above the other fields you define in this metadata set.

Next up is the metadata fields editor. Select the editor's *View* tab and select the *Fields* tab within it. Here you have all of the field types available for you to use in creating your metadata set. Icons representing the field types are listed on one side and the metadata set's canvas is on the other side. To add a field type to the metadata set, select its icon and then drag and drop it onto the canvas. The field shows on the canvas as it will for document type users. Note, by dragging a field onto a field that's already on the canvas, you can nest the new field in the existing field.

When you move your mouse over a field on the canvas, the field action icons ( ) appear. Clicking the wrench shows the field's settings table opposite the canvas. Clicking the + icon creates a duplicate of the current field and adds it below the current field. Clicking the trash can deletes the field.

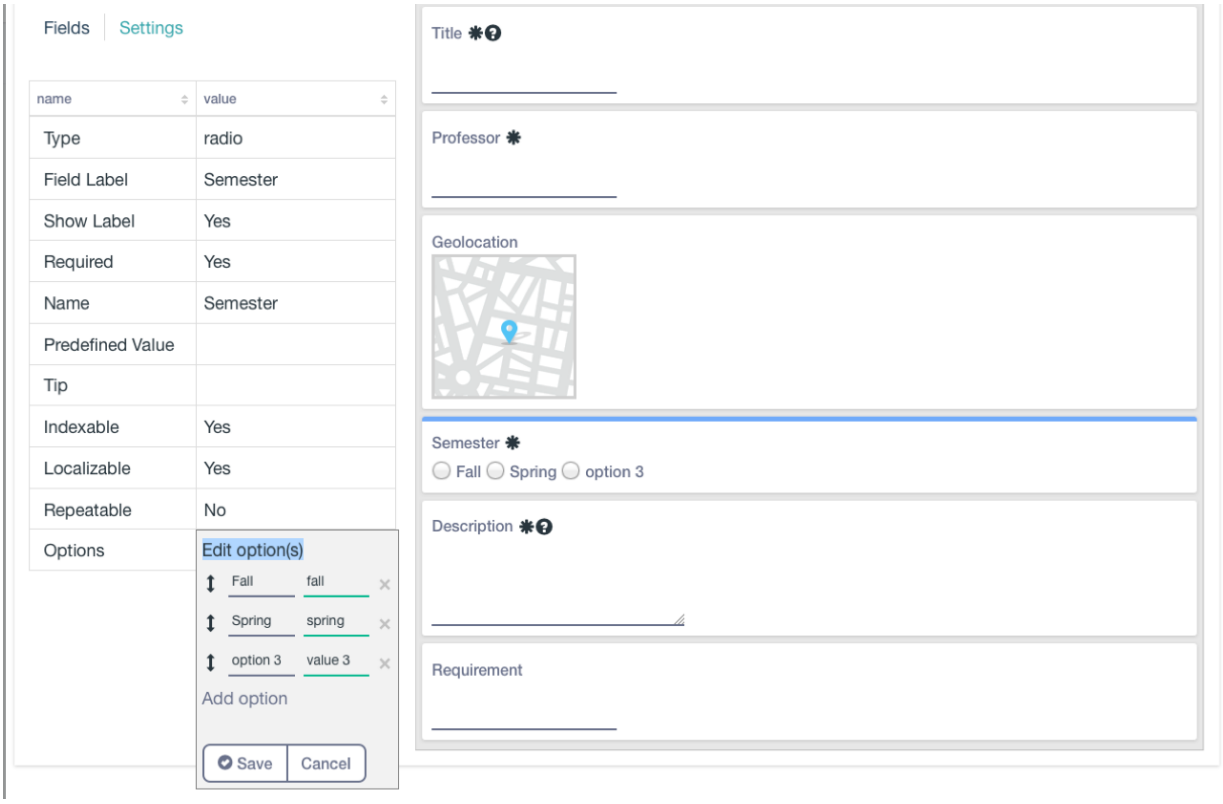


Figure 16.21: The metadata fields editor for Metadata Sets and a Document Type's Main Metadata Fields helps you define custom document types.

To configure a field's settings, just click on the field in the canvas. This automatically selects the Settings tab to show the settings table opposite the canvas. Alternatively, you can access the Settings tab by clicking the field's *Settings* icon (wrench). To edit the setting value, double-click on the setting value in the settings table and specify your new value.

Labels, default values, variable names, mouse-over tips, widths, and other settings can be configured for most fields. Some fields have a *Required* setting that lets you specify whether users must specify a value for the field. If a field's *Repeatable* setting is set to *Yes*, users can add multiple consecutive instances of the field to the document's metadata.

Click *Save* when you're done specifying your new metadata set. Now you can apply the metadata set to a new Document Type.

Creating Document Types

When apply to Document Types enrich your document library files with information and integrate them further with Liferay DXP's features. Not only do Document Types allow you to encourage, and even demand, that users specify important document metadata, but they also help you integrate the document with more of Liferay DXP's features, such as Search and Workflow. Liferay's Search picks up on Documents and Media file metadata, so users can find files faster. You can apply workflows to specific document types. And you can more cleanly organize document libraries by designating folders to exclusively hold particular document types.

Let's open up the Document Types manager to start creating a document type. In Documents and Media in Site Administration, select *Options* ( ) → *Document Types* to open the *Document Types* window. It lists all of

the current custom document types. From Search you can enter keywords to find matching document types. Selecting the check box next to *Order by* selects all of the document types. When one or more are selected the *Recycle Bin* icon ( ) appears. You can click the *Recycle Bin* icon to move the selected document types from Documents and Media to the Recycle Bin.

The following actions are available for each document type.

Edit: opens a window for modifying the document type.

Permissions: lets you grant or revoke roles the ability to update, delete, and view the document type, and configure its permissions.

Delete: moves the document type from Documents and Media to the Recycle Bin.

A document type must be associated with one or more metadata sets. When creating a new document type, you can define *Main Metadata Fields* or select *Additional Metadata Fields*. Main metadata fields are directly tied to their document type. Main metadata fields can only be created within a document type's editor window and they cannot be used with other document types. You can apply existing metadata sets to a document type in the document type's *Additional Metadata Fields* section. You can differentiate the document types that implement the same additional metadata sets by defining different main metadata fields for them.

Let's create a *Syllabus* document type to which we can apply the *Syllabus Fields* metadata set we created earlier. From the Document Types management window, click *Add*. This opens the New Document Types window. Name the document type *Syllabus*. Click the *Details* dropdown section to show a text area for entering a description.

The *Main Metadata Fields* section shows the same metadata editor that you used for defining your metadata set. The editor lets you define metadata by dragging field types from the icon area and dropping them onto the canvas. You can define a document type's metadata in the Main Metadata Fields section and/or define it using existing metadata sets.

In the *Additional Metadata Fields* section, you can select existing metadata sets to apply to the document type in addition to the document type's main metadata set. Click *Select* and choose the *Syllabus Fields* metadata set you created earlier. You've now added the metadata set to your *Syllabus* document type.

The document type's *Permissions* section lets you specify its permissions. By default, the document type is viewable by anyone, including site guests. But you can restrict the type's view, update, delete, and permissions configuration permissions to site members or the document type's owner.

When you're done creating your new document type, click *Save*. The document type is now accessible from Documents and Media's *Add* menu. When users create new instances of the document type, they're presented with metadata fields to describe the document. These fields look the same as they do in the metadata editor canvas.

Document Types and their Metadata Sets let users ascribe important information to documents. The metadata is available for users to inspect in the document's file entry view. The metadata is also searchable using Liferay DXP's Search app. As you've witnessed, Document Types and Metadata Sets make documents rich with information and integrate them more with Liferay DXP's features.

16.8 Liferay Store Types

There are several options available for configuring how Liferay's Documents and Media library stores files. Each option is a *store* which can be configured through the `portal-ext.properties` file by setting the `dl.store.impl=` property. Configuring Liferay Documents and Media stores is covered in the Document Repository Configuration Deployment Guide. Let's consider the ramifications of the various store options.

By default, Liferay Portal uses a document library store option called Simple File Store to store documents and media files on the file system (local or mounted) of the server Liferay Portal's running on. The store's

default root folder is `[Liferay Home]/data/document_library`. You can specify a different root folder from within System Settings. To access System Settings, open the *Menu* ( ) and navigate to *Control Panel* → *Configuration* → *System Settings*. From System Settings, navigate to *Platform* and then search for and select the entry *Simple File System Store*. For the store's *Root dir* value, specify a path relative to the Liferay Home or an absolute path; then click the *Update* button. The document library store switches immediately to the new root folder.

You can use a variety of methods for storing documents and media files:

Simple File System Store: uses the file system (local or a mounted share) to store files.

Advanced File System Store: nests the files into more directories by version, for faster performance and to store more files.

CMIS Store (Content Management Interoperability Services): uses a system separate from Liferay to store files.

DBStore (Database Storage): stores files in the Liferay DXP database. DBStore's file (stored as a blob) size limit is 1 gigabyte. To store such large files, use Simple File System Store or Advanced File System Store.

JCRStore (Java Content Repository): stores files to a JSR-170 compliant document repository. You can use any JCR client to access the files. The files are stored to the server's file system by default. You can optionally configure JCRStore to store files in a database.

S3Store (Amazon Simple Storage): uses Amazon's cloud-based storage solution.

Using the File System Store This is the default store. It's a simple file storage implementation that uses a local folder to store files. You can use the file system for your clustered configuration, but you'd have to make sure the folder to which you point the store can handle things like concurrent requests and file locking. For this reason, you need to use a Storage Area Network or a clustered file system.

The file system store was the first store created for Liferay DXP and is heavily bound to its database. By default, documents are stored in a `document_library` subfolder of the `data` folder in a bundle. Of course, you can change this path to anything you want by using the `dl.store.file.system.root.dir=` property.

This store creates a folder structure based on primary keys in the Liferay database. If, for example, you upload a presentation with the file name `workflow.odp` into a folder called *stuff*, the file system store creates a folder structure that looks like the figure below.

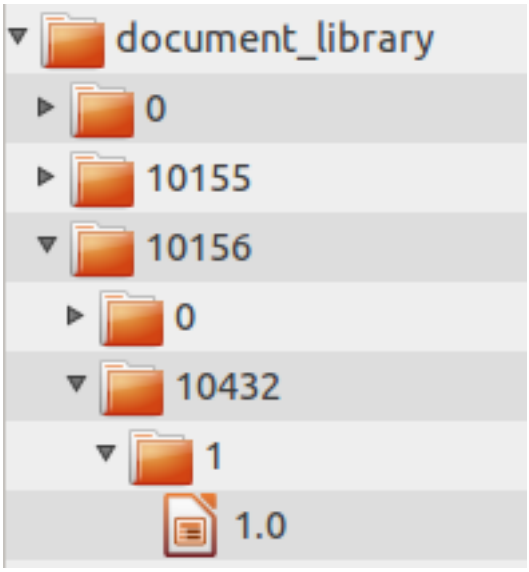


Figure 16.22: 2: Liferay's file system store creates a folder structure based on primary keys in Liferay's database.

The folder path used by Liferay for storing documents is this:

```
/companyId/folderId/numericFileEntryName/versionNumber
```

The first folder name is the company ID to which the site belongs. The second folder name is the Documents and Media folder's ID where the document resides. The third folder name is the document's numeric file entry name. Finally, the fourth name is a version number used for storing multiple versions of the document.

Note: A document's numeric file entry name is distinct from the document ID; be careful not to confuse the two! Each has an independent counter. The numeric file entry name is used in the folder path for storing the document but the document ID is not. The numeric file entry name can be found in the `name` column of the `DLFileEntry` table in Liferay's database; the document ID can be found in the `fileEntryId` column of the same table.

As you can see, the File System Store binds your documents very closely to Liferay DXP and may not be exactly what you want. If you've been using the default settings for a while and need to migrate your documents, Liferay provides a migration utility in the Control Panel → Configuration → *Server Administration* → *Data Migration*. Using this utility, you can move your documents very easily from one store implementation to another.

Speaking of other store implementations, let's look at some others Liferay provides.

Using the Advanced File System Store The advanced file system store is similar to the default file system store. Like that store, it saves files to the local file system–which, of course, could be a remote file system mount. It uses a slightly different folder structure to store files, pictured below.

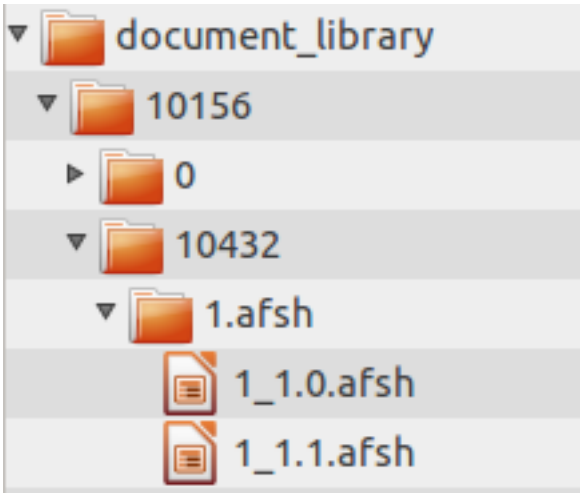


Figure 16.23: 3: The advanced file system store creates a more nested folder structure than the file system store.

So what makes the advanced file system store *advanced*? Several operating systems have limitations on the number of files that can be stored in a particular folder. The advanced file system store overcomes this limitation by programmatically creating a structure that can expand to millions of files, by alphabetically nesting the files in folders. This not only allows for more files to be stored, but also improves performance as there are fewer files stored per folder.

The same rules apply to the advanced file system store as apply to the default file system store. To cluster this, you must point the store to a network mounted file system that all the nodes can access, and that

networked file system must support concurrent requests and file locking. Otherwise, you may experience data corruption issues if two users attempt to write to the same file at the same time from two different nodes.

Follow the Deployment Guide instructions here to use the Advanced File System Store.

You may decide the advanced file system store for whatever reason doesn't serve your needs. If this is the case, you can of course mount other file systems into the documents and media library. In addition to this, you can also redefine the Liferay store to use one of three other supported protocols. We'll look at these next.

Using the CMIS Store Though you can mount as many different CMIS (Content Management Interoperability Services) repositories as you like in the Documents and Media library, you may wish also to redefine the Liferay repository to point to a CMIS repository as well. Why? Users might want to create a folder or upload content to the Liferay repository. It would be nice if that Liferay repository was connected to a clustered CMIS repository by the administrator without having to mount it through the UI. The CMIS store allows you to do just that.

Note: CMIS Store is not suitable for production use and is deprecated as of Liferay Portal CE 7.0 and Liferay DXP 7.0. Because it can have performance issues with large repositories, it's recommended that you use one of the other configurations listed above, such as Advanced File System Store, to store your Documents and Media files. This deprecation does not affect the use of external repositories. You can still connect to external repositories using CMIS.

Follow the Deployment Guide instructions here to use the CMIS Store.

The Liferay repository is connected to CMIS via the CMIS store. As long as all nodes are pointing to your CMIS repository, everything in your Liferay cluster should be fine, as the CMIS protocol prevents multiple simultaneous file access from causing data corruption.

Using the JCR Store Liferay Portal supports as a store the Java Content Repository standard. Under the hood, Liferay uses Jackrabbit, a project from Apache, as its JSR-170 compliant document repository. By default, Jackrabbit is configured to store the documents on the local file system where Liferay is installed, in the `[Liferay Home]/liferay/jackrabbit` folder. Inside this folder is Jackrabbit's configuration file, called `repository.xml`.

Note: JCR Store is deprecated as of Liferay DXP 7.0 Fix Pack 14 and Liferay Portal CE 7.0 GA4.

Using the default settings, the JCR store is not very different from the file system stores, except you can use any JCR client to access the files. You can, however, modify Jackrabbit's configuration so it stores files in a database that can be accessed by all nodes, and so that it operates as a cluster within Liferay's cluster.

Note that because of file locking issues, this isn't the best way to share Jackrabbit resources, unless you're using a networked file system that can handle concurrency and file locking. If you have two people logged in at the same time uploading content, you could encounter data corruption using this method, and because of this, we don't recommend it for a production system. Instead, if you want to use the Java Content Repository in a cluster, you should redirect Jackrabbit into your database of choice. You can use the Liferay database or another database for this purpose. This requires editing Jackrabbit's configuration file. This is covered in more detail here.

Note that this configuration doesn't perform as well as the advanced file system store, because you're storing documents in a database instead of on the file system. But it does have the benefit of clustering well.

Using Amazon Simple Storage Service Amazon's simple storage service (S3) is a cloud-based storage solution that you can use with Liferay DXP. All you need is an account, and you can store your documents to the cloud from all nodes, seamlessly.

When you sign up for the service, Amazon assigns you unique keys that link you to your account. In Amazon's interface, you can create "buckets" of data optimized by region. Once you've created these to your specifications, follow the instructions found here to connect your repository to Liferay DXP.

Consult the Amazon Simple Storage documentation for additional details on using Amazon's service.

Chapter 17

Using a SharePoint Repository

The Liferay Marketplace app *Liferay Connector for SharePoint* lets users add a SharePoint repository type to Liferay DXP's Documents and Media Library. It supports SharePoint 2013 and SharePoint 2016.

Note: To access SharePoint 2010, you must use the Liferay Marketplace app *Liferay Connector for SharePoint 2010*.

Note: Liferay Connector for SharePoint uses Azure ACS with OAuth 2 for SharePoint server authorization. You must therefore enable HTTPS support in your app server. Consult your app server's documentation for instructions. For example, the required steps for Tomcat can be found in its documentation.

Liferay Connector for SharePoint provides these key benefits:

- Reading/writing documents and folders
- Document check-in, check-out, and undo check-out
- Downloading documents
- Moving folders and documents within the repository
- Getting revision history
- Reverting to a revision

The app uses SharePoint's API, which has these limitations:

- Version history is lost when moving or renaming a file without first checking it out.
- You can't change file extensions, you can only change file names.
- A file's current name propagates to all previous versions.
- The user who checks out a file is the only one who can see the version number of that file's working copy.
- Queries for suffixes or intermediate wildcards convert to queries for containment.
- Comments, ratings, and using a SharePoint folder as a Documents and Media root folder are unsupported.

To use a SharePoint repository in Documents and Media, you must first create an application in SharePoint and authorize it to access the repository. The guides in this section walk you through this process.

17.1 Creating a SharePoint Application

The steps covered in this section apply to SharePoint Online. If you're using SharePoint On-Premises, please consult your administrator for the requirements.

Follow these steps to register Liferay DXP as an application in a SharePoint instance:

1. Go to your SharePoint installation's URL: `https://[your-site-name].sharepoint.com/_layouts/15/appregnew.aspx`.

2. Provide the information below for your app:

   - **Title:** The name displayed in Documents and Media.
   - **Domain Name:** The application's domain name along with the port (e.g., `localhost:8228`)
   - **Redirect URL:** The application's URL. The URL must use HTTPS.

3. Click the two *Generate* buttons to generate a client ID and client secret for Liferay DXP.

   Here's an example configuration:

   - **Client ID:** `1234a56b-7890-1234-5ccc-67d8ea9b0c1c`
   - **Client Secret:** `1ABCDEfGh2IJKLmNoP3QrStuvwX41YzAB+CDEFg20G3=`
   - **Title:** `My Application's Title`
   - **App Domain:** `localhost:8228`
   - **Redirect URL:** `https://localhost:8228/c/document_library/sharepoint/oauth2`

4. Now you must grant Liferay DXP write and search permissions over the SharePoint instance. Other permissions are ignored. Go to `https://[your-site-name].sharepoint.com/_layouts/15/appinv.aspx`.

   In the *APP ID* field, enter the Client ID of the application you just created and click *Search*. Consult Microsoft's documentation for details on how to configure the Permission Request XML. Here's an example configuration that grants the application write and search permissions over the SharePoint instance:

   ```
   <AppPermissionRequests>
       <AppPermissionRequest
           Scope="http://sharepoint/content/sitecollection/web/list"
           Right="Write"
       />
       <AppPermissionRequest
           Scope="http://sharepoint/search"
           Right="QueryAsUserIgnoreAppPrincipal"
       />
   </AppPermissionRequests>
   ```

5. Once you provide the permissions XML, click *Create*.

6. Next, go to *Settings* → *Site App Permissions* in your SharePoint installation. You can also access this page directly with a URL similar to this: `https://[your-site-name].sharepoint.com/_layouts/15/appprincipals.aspx?Scope=We` Find your application in the registration list and note its *app identifier*. You'll use this to create a repository configuration. Here's an example app identifier:

   ```
   i:0i.t|ms.sp.ext|6123d38d-2998-4972-9aaa-71a4da9f3a5a@b9c24ab3-ad34-4943-ab57-729d8c329053
   ```

Great! Now Liferay DXP is registered as an application in your SharePoint instance. Next, you must create a new SharePoint repository type.

17.2 Creating a New SharePoint Repository Configuration

To connect to a remote SharePoint server you must create a repository configuration. To do so, follow these steps:

1. In Liferay DXP, open the Control Panel and go to *Configuration* → *System Settings* → *Collaboration* → *Sharepoint OAuth2*.

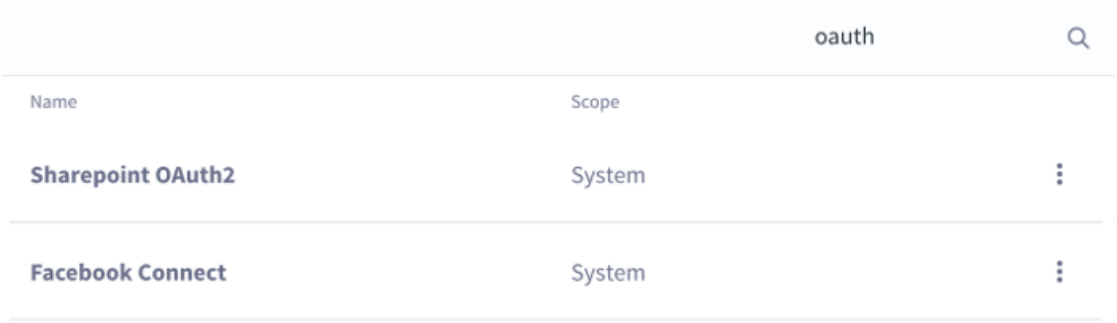


Figure 17.1: Use the *Sharepoint OAuth2* system setting to create a new SharePoint repository configuration.

2. Click the *Add* icon (+) to begin creating a new configuration.
3. In the new Repository form, specify values for the following fields (your SharePoint administrator can provide you with this information):

   - **Name:** The configuration's name.
   - **Authorization grant endpoint:** The URL used to request OAuth2 authorization grants. The URL follows this pattern in SharePoint Online: `https://[your-site-name]/sharepoint.com/_layouts/oauthauthorize.as`
   - **Authorization token endpoint:** The ACS URL. In SharePoint Online with Azure ACS, the URL follows this pattern: `https://accounts.accesscontrol.windows.net/[App ID]/tokens/OAuth/2`.
   - **Client ID:** The client ID.
   - **Client Secret:** The client secret.
   - **Scope:** The permission set required for your tokens. Valid scopes are configured during your app's SharePoint registration.
   - **Tenant ID:** The Tenant ID. If you're using SharePoint Online, you can use the same App ID you embedded in the *Authorization token endpoint*.
   - **Site domain:** The site domain registered for your application.
   - **Resource:** This value depends on the ACS service you use. In SharePoint Online with Azure ACS, the value follows this pattern: `00000003-0000-0ff1-ce00-000000000000/[your-site-name].sharepoint.com@[tenant ID]`.

4. Click *Save*.

Awesome! Now that your SharePoint repository is configured, you can mount it in Documents and Media Library.

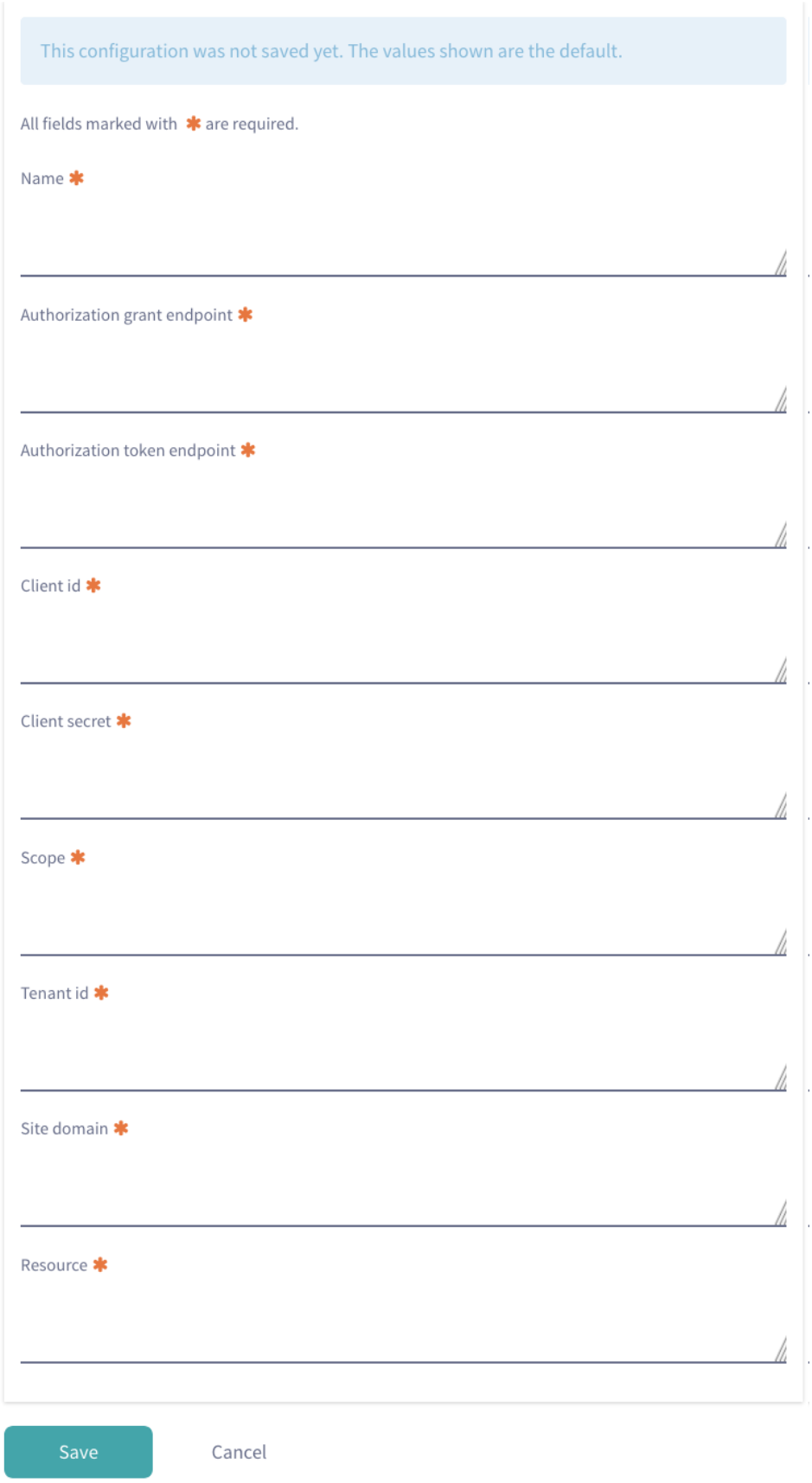


Figure 17.2: The New Repository form is where you specify access to the remote SharePoint server.

17.3 Adding a SharePoint Documents and Media Repository

Follow these steps to add your SharePoint Library to Documents and Media:

1. Add the *Documents and Media* application to a page, if you haven't already.

2. Inside the Documents and Media application, click the *Add* icon (+) and select *Repository*. The New Repository screen appears.

3. Select the repository type for the SharePoint OAuth2 configuration you created. For example, if your configuration is named `Foo`, the repository type is listed as `SharePoint (Foo)`.

4. Specify values for these fields:

    - **Site Absolute URL:** Resolves relative URLs. For SharePoint Online, the value follows this pattern: `https://[your-site-name].sharepoint.com`.
    - **Library Path:** A relative path from the *Site Absolute URL* that points to the SharePoint Document Library you want to mount in Documents and Media (an example path could be `Shared Documents`).

5. Click *Save*.

The Documents and Media Library now lists your SharePoint repository.

Note: The first time you access a mounted SharePoint repository, you must provide login credentials and also grant permission for Liferay DXP to access the remote SharePoint repository.

Sweet! Now that you've added a SharePoint Repository to Documents and Media, you can access and modify SharePoint Library files in your Liferay DXP instance.

17.4 Using External Repositories

Liferay DXP's Documents and Media lets you store files in a number of different ways and lets you access external repositories in addition to the built in document repository. Your administrator can learn how to set options for storing the Documents and Media Library's files here. To connect the Documents and Media library to external repositories, read on.

Documents and Media allows users to connect to multiple third-party repositories. Some of the features supported with third-party repositories include:

- Reading/writing documents and folders
- Document check-in, check-out, and undo check-out
- Downloading documents
- Moving folders and documents within the repository
- Getting revision history
- Reverting to a revision

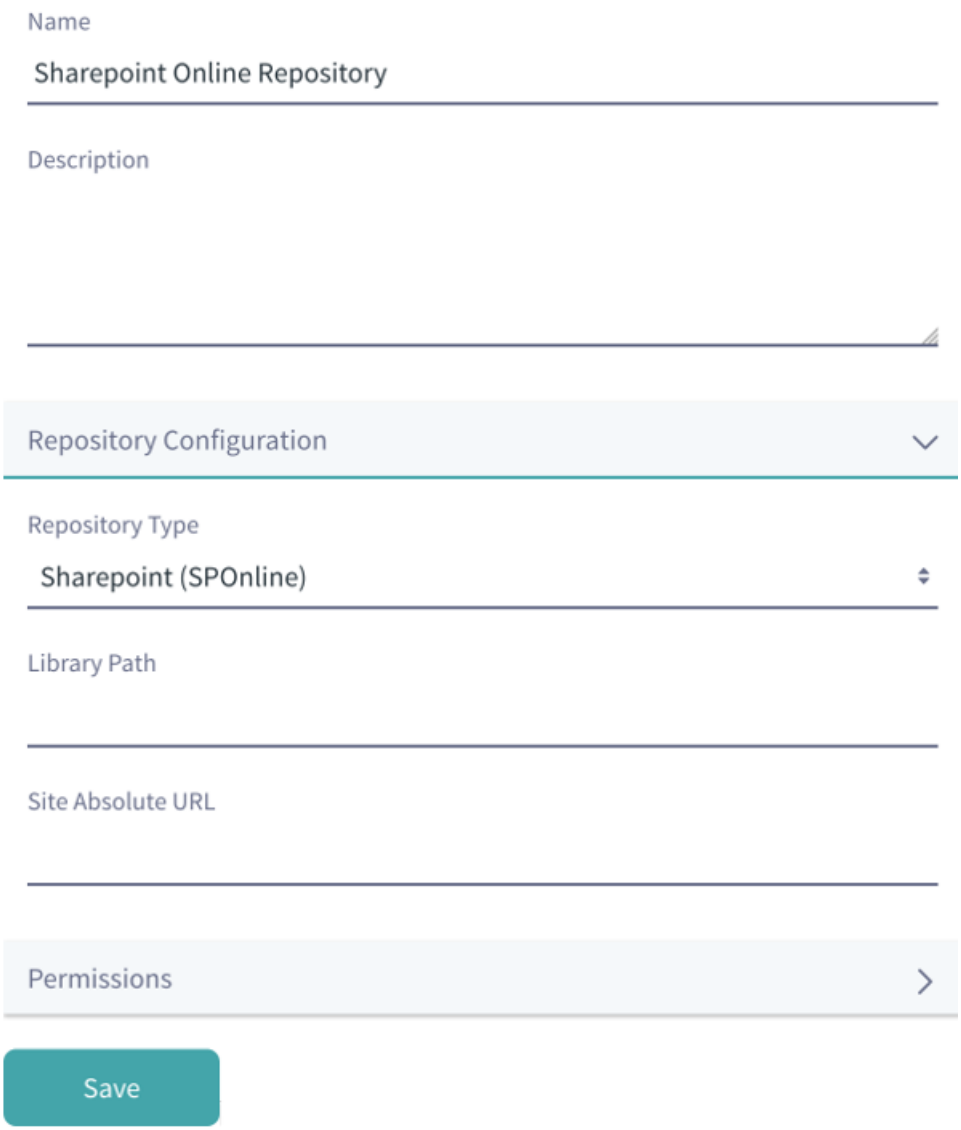


Figure 17.3: The Repository Configuration form is where you specify access to the SharePoint Library you want to use.

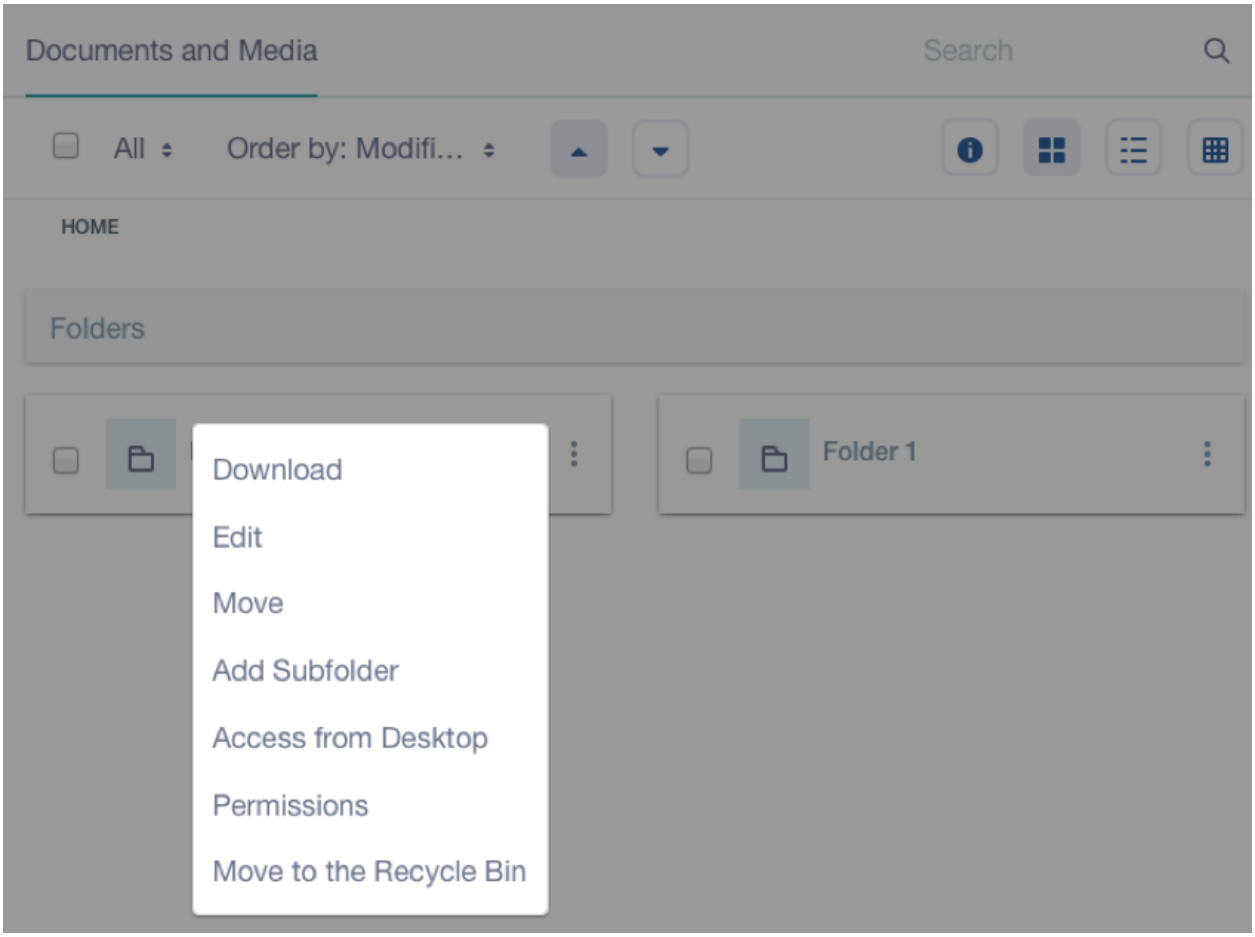


Figure 17.4: In Documents and Media, external repositories look similar to folders. You can perform supported operations on their files. This figure shows an Alfresco CMIS repository named *My Alfresco*.

Liferay DXP's Documents and Media can inter-operate with repositories that support the Content Management Interoperability Services (CMIS) standard. CMIS defines a model for interacting with repository files. Liferay DXP supports interoperability through CMIS version 1.1 (the current CMIS version) and earlier. Liferay DXP can access CMIS repositories using AtomPub and Web Services protocols.

Here are the CMIS repository integration steps:

1. Adjust portal properties
2. Synchronize user accounts between Liferay DXP and the repository
3. Add the repository to Documents and Media

Interoperability with SharePoint and Documentum repositories is available to EE subscribers with the Liferay SharePoint Connector and Liferay Documentum Connector apps, available on the Marketplace.

Let's start with setting portal properties.

Step 1: Adjust Portal Properties

In order to authenticate with the third-party repository, you need to store passwords for the user sessions. You must configure an authentication type that supports storing passwords to the user sessions.

Important: Since authentication with single sign-on (SSO) does not store encrypted passwords in the user sessions, SSO can't be used with the external repository types.

Let's configure Liferay Portal for what's required in authentication. In a `portal-ext.properties` file in your Liferay Home, set a `session.store.password` portal property to `true`:

```
session.store.password=true
```

Next, make sure to authenticate the same way on both Liferay DXP and the repository. You can do so by authenticating based on screen name. Set the following `company.security.auth.type` portal property:

```
company.security.auth.type=screenName
```

Alternatively, you can configure both of these properties in the Control Panel under *Instance Settings* → *Authentication*.

Step 2: Synchronize the User Accounts

External repository integration requires Liferay DXP users use the same screen name and password as their external repository user. For details on adding and managing users, refer to User Mangement. Make sure to sign in to Liferay DXP as such a user whenever you're accessing the repository from it.

Important: Use lowercase for user names in the external repository, as Liferay screen names are lowercase.

Note: In the external repository, make sure to associate users with necessary roles for accessing repository files. In Nuxeo, for example, a user must belong to the Nuxeo *Administrators* group to access Nuxeo repository files via CMIS.

Adding a *Repository* mapped to the external repository is the final step.

Step 3: Add the Repository to Documents and Media

The last step is adding a to Documents and Media a *Repository* to inter-operate with the external repository. Documents and Media can connect with a CMIS repository using the Atom Publishing (AtomPub) protocol or Web Services protocol. Let's explore accessing a CMIS Repository using AtomPub.

Adding a CMIS Repository that Uses AtomPub AtomPub is the easiest protocol to specify for communicating with a CMIS repository.

Follow these steps to access a repository with CMIS AtomPub:

1. Sign in to Liferay DXP as a user that also exists in the external repository and can access files you want to use in that repository.

2. In Documents and Media *Home*, click the *Add* icon () and select *Repository*. The *New Repository* screen appears.

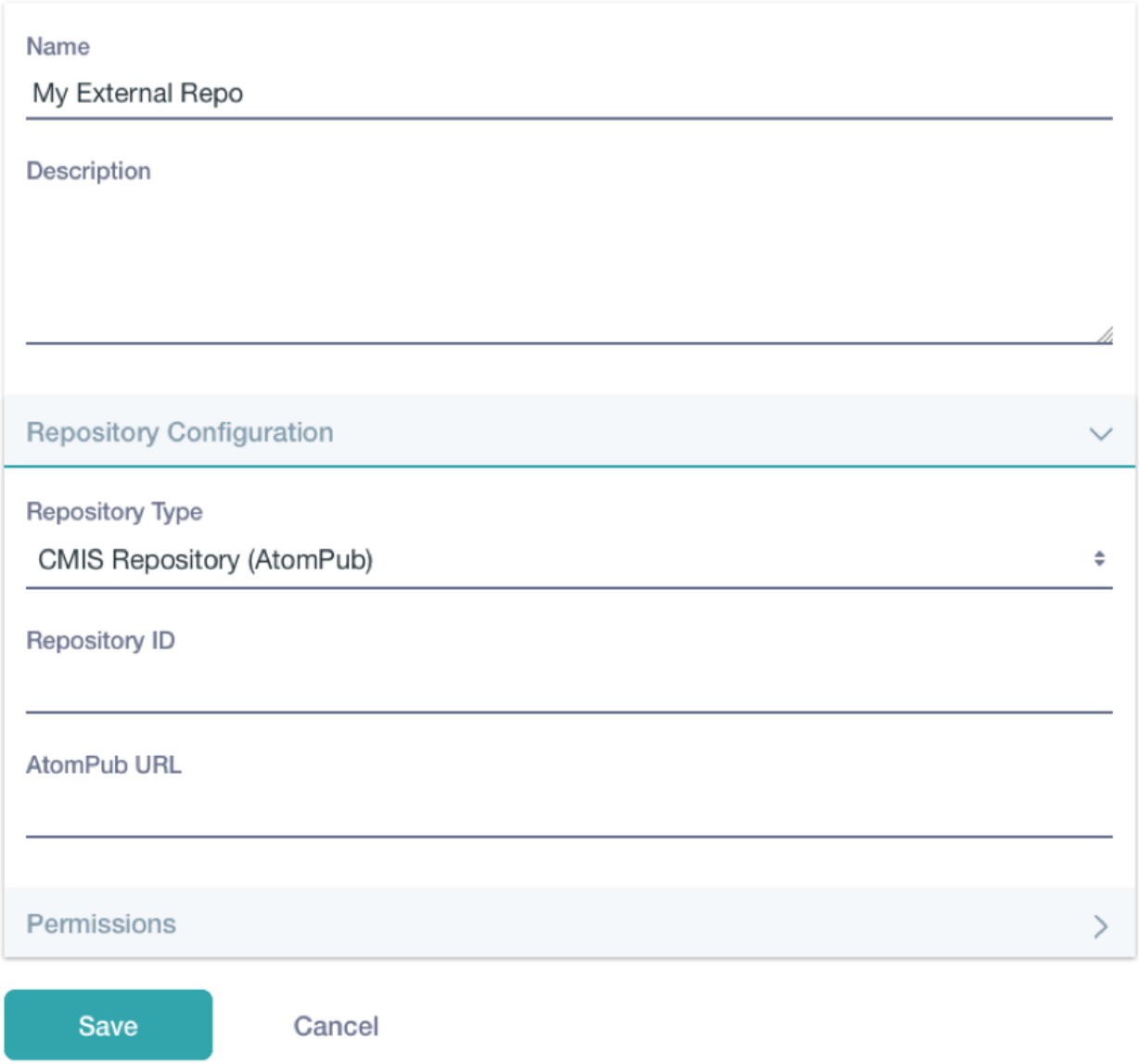


Figure 17.5: Here's the form for accessing a CMIS repository using AtomPub.

3. Enter an arbitrary *Name* for the repository and optionally enter a *Description*.

4. Click on the *Repository Configuration* section to access its form. Specify values for the following fields:

   - **Repository Type**: *CMIS Repository (AtomPub)*
   - **Repository ID**: Leave this blank, as Liferay DXP generates the ID automatically
   - **AtomPub URL**: Enter the AtomPub URL for CMIS per the repository's documentation. Example URLs are listed below. If you copy them, make sure to replace the `[host]` and `[port]` values with your own.
     - Alfresco 5.1 CMIS 1.0: `http://[host]:[port]/alfresco/cmisatom`
     - Alfresco 5.1 CMIS 1.1: `http://[host]:[port]/alfresco/api/-default-/public/cmis/versions/1.1/atom`
     - Nuxeo 7.10: `http://[host]:[port]/nuxeo/atom/cmis`

5. Click *Save*

The external repository appears in Documents and Media. You can navigate the repository and manage files.

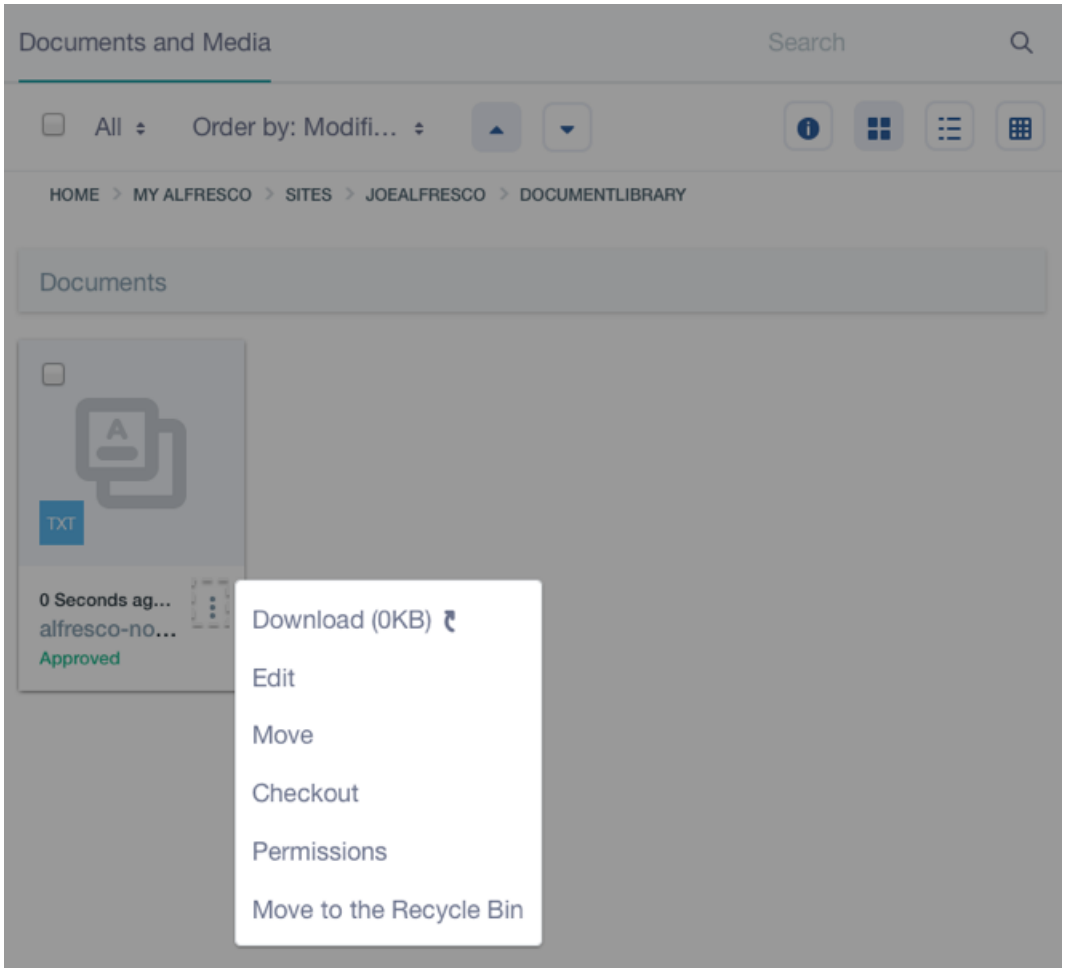


Figure 17.6: Documents and Media lets you perform CMIS supported actions on external repository files.

Next, let's access a CMIS repository with Web Services.

Adding a CMIS Repository that Uses Web Services After adjusting your portal properties and adding any user accounts required by the repository, you can add a Documents and Media Repository that inter-operates with an external repository that supports CMIS with Web Services.

Follow these steps to add such a repository:

1. In Documents and Media *Home*, click the *Add* button and select *Repository*. The *New Repository* screen appears.

2. Enter an arbitrary *Name* for the repository and optionally enter a *Description*.

3. Click on the *Repository Configuration* section to access its form

4. For **Repository Type:**, select *CMIS Repository (Web Services)*.

5. Enter the web service URLs (WSDL URLs) per your CMIS repository's documentation. Example URLs are listed after these steps. Make sure to enter **URLs** for the following services:

    - Web Services ACL Service
    - Web Services Discovery Service
    - Web Services Multi-Filing Service
    - Web Services Navigation Service
    - Web Services Object Service
    - Web Services Policy Service
    - Web Services Relationship Service
    - Web Services Repository Service
    - Web Services Versioning Service

```
Tips: In a browser, you can open the repository's WSDL  file and verify each
service in it. Also make sure to specify the entire URLs--they all end with
`?wsdl`.
```

```
![ Here's an example of the web service URLs for an Alfresco repository.](./images/dm-repo-types-alfresco-ws.png)
```

6. Leave *Repository ID* blank, as Liferay DXP generates the ID automatically.

7. Click *Save*

Your repository appears in Documents and Media. You can navigate into the repository and work with files as you do other files in Documents and Media.

Liferay DXP's CMIS repository integration gives you flexibility to access external repository files in addition to your existing Documents and Media files.

17.5 Administering Liferay Sync

Liferay Sync is an add-on product for Liferay DXP that synchronizes files between your Liferay DXP server and users' desktop and mobile environments. With Liferay Sync, users can publish and access shared documents and files from their native environments without using a browser. Clients for Windows and Mac OS desktops and Android and iOS mobile platforms are supported. As users add and collaborate on documents and files, Liferay Sync automatically synchronizes them across all configured Sync clients and in your Liferay DXP installation. Liferay Sync integrates completely with Liferay DXP so that features such as authentication and versioning function in the supported environments. The Liferay Sync desktop client stores files locally so they're always available, even when users are offline, and when they go back online, it automatically synchronizes files upon client reconnection. The Liferay Sync mobile client saves storage space on users' devices by downloading only the files they choose.

This guide walks you through setting up Sync for your Liferay DXP installation and its sites. In addition to basic configuration, important topics such as security and accidental data loss are covered. Note, you must be an administrator to set up your Liferay DXP installation for Sync. If you're not an instance or site administrator, you can skip this guide and move on to installing and using the Sync desktop client.

To prepare your Liferay DXP installation for Sync, you must first install Sync's prerequisite apps.

Installing Sync's Prerequisites

Liferay Sync requires that your Liferay DXP installation contains the most recent versions of the following Liferay Marketplace apps:

- Liferay IP Geocoder: Uses geographic IP lookup to get the approximate location of devices that register with Sync. This is a security measure that lets you see where the devices are that connect to your Liferay DXP installation via Sync.
- Liferay OAuth Provider: Enables OAuth in your Liferay DXP installation. OAuth is required to use Sync with SSO (single sign-on). The next section details additional steps that are required to use Sync with SSO.
- Liferay Sync Connector: Lets you enable and configure Sync in your Liferay DXP installation. For example, you can disable Sync across the instance or on a site-by-site basis. Note that Sync is enabled by default for all your Liferay DXP installation's sites.

These apps are preinstalled in the latest Liferay DXP releases. To verify that you have them, navigate to *Control Panel → Apps → App Manager*. You can search for them in the *Search* field at the upper right. If your Liferay DXP installation doesn't contain these apps, install them from Liferay Marketplace. Click here for instructions on installing apps from Liferay Marketplace.

Note that older Sync 1.x clients can't connect to the latest Sync Connector app.

Note: The required Liferay Sync Security module is included and enabled by default in Liferay DXP. You can verify this by ensuring that the `SYNC_DEFAULT` and `SYNC_TOKEN` entries are enabled in *Control Panel → Configuration → Service Access Policy*.

If you want to use Sync Connector's default settings and are fine with Sync being enabled for all your Liferay DXP installation's sites, you can skip the next section. Before directing your users to install and configure the Sync desktop and mobile clients, however, **Make sure to read** this guide's sections on preventing

accidental file deletion and ensuring Sync security. You should also **warn your users** about the potential for accidental data loss.

Next, you'll learn how to configure Sync for SSO. You can skip this section if your Liferay DXP installation doesn't use an SSO server.

Configuring Sync to Use SSO

If your Liferay DXP installation uses an SSO (single sign-on) server, you must ensure that Sync can access the following URLs without being redirected to your SSO server. Sync can't work without direct access to these URLs. You must therefore whitelist these URLs:

```
http(s)://<portal-address>/c/portal/oauth/*
http(s)://<portal-address>/api/jsonws/sync-web.*
http(s)://<portal-address>/sync-web/*
```

For example, if your Liferay DXP installation's address is `https://www.joesblog.com`, then you must whitelist the following URLs:

```
https://www.joesblog.com/c/portal/oauth/*
https://www.joesblog.com/api/jsonws/sync-web.*
https://www.joesblog.com/sync-web/*
```

Sync uses the paths specified in the first URL for communication via OAuth, and the paths specified in the remaining URLs for normal communication with your Liferay DXP installation.

You must also enable OAuth in the Sync Connector app. The next section covers this, as well as other information on how to configure Sync.

Configuring Liferay Sync

Sync Connector lets you manage how or if clients connect to your Liferay DXP installation. You can also configure default file permissions on a per-site basis and manage the devices that connect to your Liferay DXP installation. To access Sync Connector, select *Control Panel → Configuration → Sync Connector Admin*.

Sync Connector Admin has three tabs:

1. **Settings:** Control Sync's general behavior. These settings apply globally to Sync.

    - **Allow the use of Sync?:** Whether Sync is enabled.
    - **Allow users to sync their personal sites?:** Whether users can sync data with their personal sites.
    - **OAuth Enabled:** Whether clients use OAuth and any configured SSO autologin filters for authentication.
    - **Allow LAN Syncing?:** Whether desktop clients attempt to download updates from other desktop clients on the same local network before downloading from the server. This can help reduce server load and increase data transfer speeds. Note that LAN syncing only works with clients that also enable it.
    - **Max Connections:** The maximum number of simultaneous connections each client is allowed per account. For example, if Max Connections is three, a client can simultaneously upload or download up to three files for each account. Note, this setting operates on a per client basis. If Max Connections is set to three and a user has two clients connected to an account (which is possible if Sync is installed on two different devices), then the user is effectively allowed six simultaneous connections. While increasing Max Connections can speed up file transfers it also places a heavier load on the server. *Max Connections* is set to one by default.

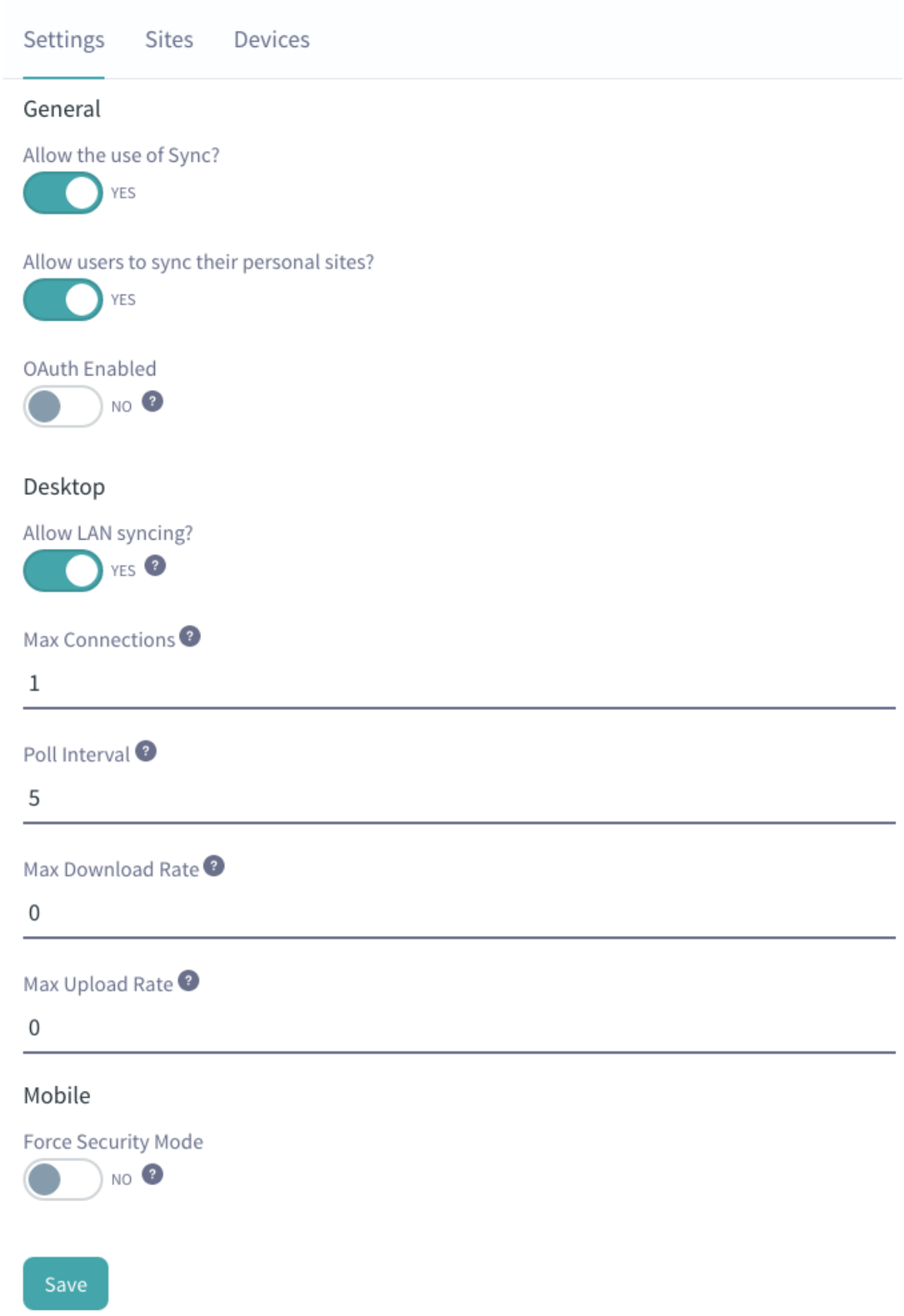


Figure 17.7: The Control Panel's Configuration section contains Sync Connector Admin.

- **Poll Interval:** The frequency in seconds that clients automatically check the server for updates. For example, if set to ten, connected clients check the server for updates every ten seconds. The default Poll Interval is five.
- **Max Download Rate:** The maximum transfer rate, in bytes, at which clients can download. A value of `0` specifies no limit. This setting takes precedence over clients' download rate setting.
- **Max Upload Rate:** The maximum transfer rate, in bytes, at which clients can upload. A value of `0` specifies no limit. This setting takes precedence over clients' upload rate setting.
- **Force Security Mode:** Whether to force security mode on mobile clients. Security mode encrypts Sync files on the device and requires a passcode when accessing the Sync mobile app.

2. **Sites:** Control Sync on a per-site basis.

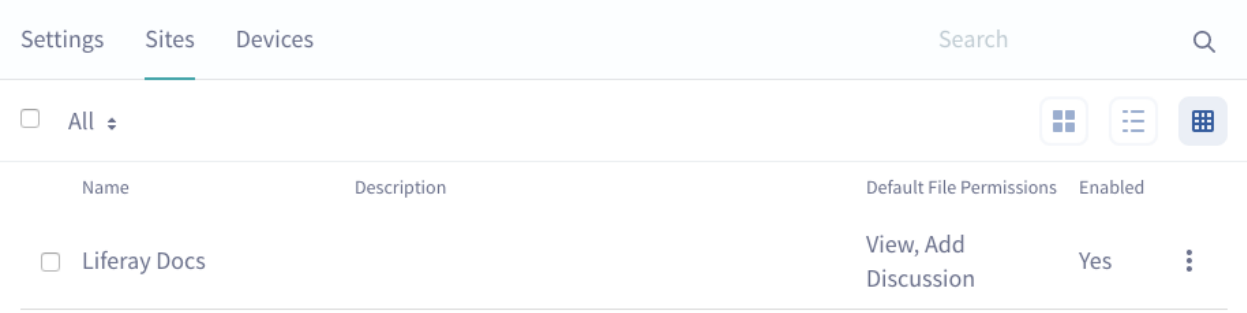


Figure 17.8: Sync Connector Admin's Sites tab lets you manage Sync on a per-site basis.

For each site in the Liferay DXP installation, the Sites tab lists each site's default file permissions (more on this in a moment) and whether Sync is enabled for that site. Sync is enabled by default for all sites. To disable Sync for a site, click the site's *Actions* button (⋮) and select *Disable Sync Site*. To disable multiple sites at once, select their checkboxes and click the *Disable Sync Sites* link that appears above the table. **Please use caution** when disabling Sync for a site, as doing so **deletes** files for that site from the Sync clients. Disabling Sync for a site, however, doesn't affect the site's files in the Liferay DXP installation.

```
**Warning:** Disabling Sync for specific sites from Sync Connector Admin can
result in data loss across clients. If Sync is disabled for a site users are
currently syncing, any files in the clients' sync folders for that site are
automatically deleted from their clients. If a user is offline when Sync is
disabled for a site, any offline changes or additions they make are deleted
upon client reconnection.
```

```
You can enable Sync for a site by selecting *Enable Sync Site* from its
Actions button. To enable Sync for multiple sites, select their checkboxes
and click the *Enable Sync Sites* link that appears above the table.

Make sure that each site for which Sync is enabled has a Documents and Media
app on at least one of its pages. If a site doesn't have the app on any of
its pages and users click the *Open Website* link from their Sync menus, the
error message *The requested resource was not found* appears.
```

```
The Sites tab also lets you set default file permissions for files uploaded
from Sync clients. The process for setting permissions is nearly the same as
for enabling or disabling Sync for sites. To set the default file
permissions for a single site, click its Actions button and select *Default
File Permissions*. This opens a window that lets you select the default file
permissions for that site. Click *Choose* for the permissions you want to
use.

![ Click *Choose* to select the default file permissions for a site in Sync.](./images-dxp/sync-admin-03.png)

To set the default file permissions for several sites, select the checkboxes
for the sites, click the *Default File Permissions* link that appears above
the table, and select the permissions you want to use. Default file
permissions might behave differently than you'd expect. They control *only*
the permissions for new files uploaded through the Sync clients; they don't
affect permissions for uploading or restrict document owners (the user who
originally uploaded a document) in any way. For example, even if you set a
site's default file permissions to View Only, that site's users can still
upload new documents to the site. The file's owner has edit permission; the
rest of the site's users have the View Only permission.
```

3. **Devices:** View and manage the devices registered with Sync.

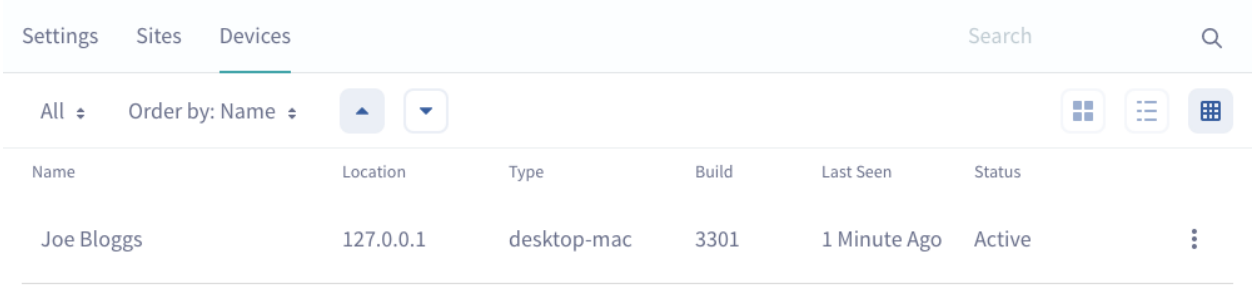


Figure 17.9: Sync Connector Admin's Devices tab lists all the devices Sync has registered for the Liferay DXP installation.

Each row in the Devices tab's table represents a device. The *Name* column lists the user that registered the device. The remaining columns list each device's location, client type, client build number, last connection date, and status. Each device's Actions button (⋮) lets you manage that device. You can change a device's status from Active to Inactive by selecting *Actions* → *Disable Sync Device*. Inactive devices can't sync with the Liferay DXP installation. Inactive mobile devices also can't access local Sync files. Once a device is Inactive, you can erase Sync files from it by selecting *Actions* → *Wipe Sync Device*. This also signs the device out and removes the account from the client. If the device is offline, this happens when it tries to reconnect. The Actions menu also lets you enable or delete an Inactive device. Deleting a device only removes it from the list of registered devices; it can still reconnect and reregister.

Awesome! Now that you know how to configure Sync in your Liferay DXP installation, it's time to look closer at a use case that warrants special attention for administrators: accidental file deletion.

Protecting Against Accidental File Deletion

Liferay Sync's power rests in its ability to propagate between the server and connected Sync clients. When a user deletes a file from a connected client, Sync also deletes the file on the server and in any other connected

clients. Likewise, if a user deletes a file on the server, Sync also deletes the file in all connected clients. In other words, anywhere a user deletes a file, Sync deletes it *everywhere*. You're probably thinking, "This is a disaster waiting to happen!" Don't fret! Liferay DXP's Recycle Bin is enabled by default and lets you recover deleted files. You can access the Recycle Bin from each site's *Site Administration* menu.

Warning: Liferay Sync automatically propagates file and folder deletion through the Liferay DXP installation and in all connected clients. If an instance or site administrator disables the Recycle Bin, deleted files can't be recovered.

Liferay DXP instance and site administrators can, of course, disable the Recycle Bin. Disabling the Recycle Bin in a site, however, leaves the site vulnerable to accidental file deletions that propagate through Sync.

Ensuring Sync Security

As an administrator, you're undoubtedly concerned about the security of all connections to and from your Liferay DXP installation. As long as your server is configured to use HTTPS, Sync clients communicate securely with your instance with user-supplied credentials. Users can only access the documents and sites they're permitted to access. To support Security Mode in the Sync mobile client and securely transmit files, your Liferay DXP server must also use SSL. The next section demonstrates how Sync's permissions work with your Liferay DXP installation's permissions.

Liferay Sync Permissions Demonstration

Sync uses Liferay DXP's default permissions to determine files and folders to sync with the user's devices. It can only sync files a user can access in the Liferay DXP instance. After installing the desktop Sync client, you can follow the steps below to test this functionality.

First, enter the text `classified information` into a new text file and save it on your desktop as `secret.txt`. Then use your browser to sign into your Liferay DXP installation and create a new user with the user name *secretagent* and the email address *secretagent@example.com*. Give this user a password and then create a new private site called *Secret Site*. Create a page on the site and add the Documents and Media app to it. Then add the secretagent user to the Secret Site and grant the *Site Administrator* role to the user. Log in as secretagent and navigate to the Secret Site. Then upload the `secret.txt` document to the Documents and Media app. Make sure you also have a user that isn't a member of the Secret Site and therefore doesn't have access to any of its documents through Sync. If you don't have such a user, create one now.

Next, configure your Liferay Sync client to sign in with the secretagent user's credentials and sync with the Secret Site. Open the Liferay Sync menu from the system tray and select *Preferences*. In the *Accounts* tab, click the plus icon at the window's bottom left to add an account. Provide the secretagent user's credentials and uncheck all Liferay DXP sites except the Secret Site. Now confirm that Sync downloaded the `secret.txt` file to your new Sync folder. Open it and check that it contains the text `classified information`. Next, use Sync to connect to your Liferay DXP installation with the user that doesn't belong to the Secret Site. The file doesn't sync because this user isn't a site member.

Now go to Sync Connector Admin and set the Secret Site's default file permissions to View Only. Create a new user, add it to the Secret Site, and add its account in your Liferay Sync client. As with the secretagent user, Sync downloads the `secret.txt` file to this user's local Sync folder because the user is a member of the Secret Site. Now edit and save this file. Even though you can edit and save it locally, the edits aren't synced because the site's default file permissions are View Only. After attempting the sync, a red *x* appears next

to the file in the local Sync folder. Right click the file to see the error. It confirms the user doesn't have the required permissions.

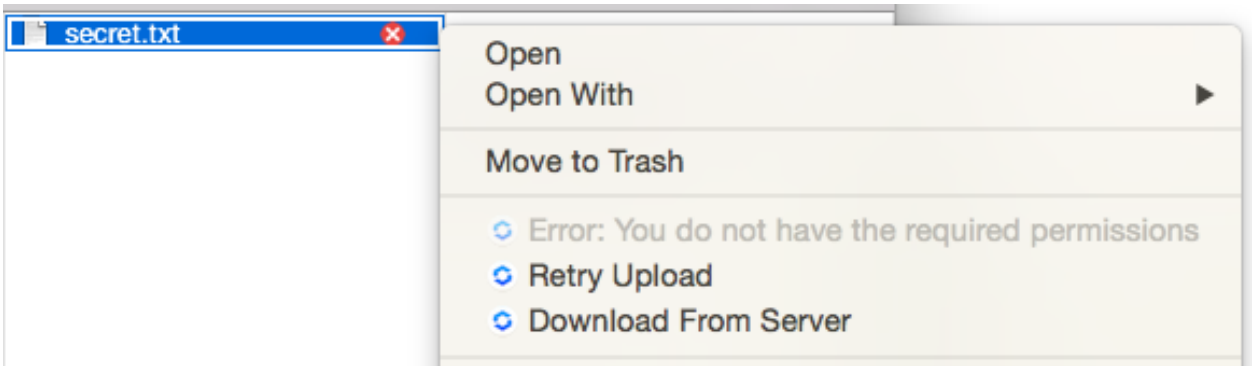


Figure 17.10: The upload error occurs because the user only has permission to view files.

To confirm that the error didn't propagate through Sync, open the file in the secretagent user's local Sync folder. It still contains the original text. Likewise, the original file remains in the site's Documents and Media portlet. To get rid of the error in the other user's local Sync folder, return there and then right click the file and select *Download From Server*. This replaces the file with the latest file in the Liferay DXP installation.

Now edit `secret.txt` in the secretagent user's local Sync folder. When you check the file in the other user's local Sync folder and on the server, notice that Sync propagated the edits. The changes were propagated because the secretagent user owns the file in the instance. Owners can do anything with their files, even when the site's default file permissions are set to View Only.

Congratulations! You've successfully set up a Liferay Sync folder that can only be accessed by site members and administrators. You've also seen how Sync's default file permissions work. By using Liferay DXP's permissions alongside Sync Admin's default file permissions, Sync gives administrators two levels of control over files in their sites.

Great! Now you know how to enable and configure Liferay Sync in your Liferay DXP instance. The next article explains how end users can install and configure the Sync desktop client.

17.6 Using Liferay Sync on Your Desktop

Liferay Sync synchronizes files between your Liferay DXP sites and desktop devices. It lets you work with your files without using a browser. The Sync clients also ensure that the files are updated with the latest changes made by other users. To use Liferay Sync in your desktop environment, you must install the Sync desktop client. It's currently available for Windows and Mac OS. The Sync client stores files locally so that they're always available, even when you're offline. Files are automatically synchronized upon your client's reconnection to your Liferay DXP instance.

On your desktop devices, Liferay Sync creates a new folder structure that it uses to synchronize files. You can treat the files the same as you do any ordinary file. Credentials, Sync folder location, and other options are configured in the client. Also, native desktop notification events inform you of what Sync is doing. The native menu and task bar integration keep Sync controls within easy reach.

This guide walks you through setting up and using the Liferay Sync client on your desktop. Before proceeding, check with your Liferay DXP instance or site administrator to ensure that Sync is enabled for your sites. You then need to install the Sync client on your desktop device. The next section walks you through installing the client.

Installing the Desktop Liferay Sync Client

You can download the desktop client from the Liferay Sync downloads page. Note that you'll need a Liferay account for this. Once you've downloaded the appropriate desktop client for your operating system, installing Liferay Sync on Windows or Mac OS is straightforward.

To install the Liferay Sync client on Windows, you must have administrator privileges. Upon launching the Windows application installer, you're prompted to choose an install location. Select an appropriate location and click *Install*. Sync automatically starts after the installation finishes. The first time Sync runs, you need to configure it to connect and sync with your Liferay DXP instance. The configuration steps are shown below, after the Mac installation instructions.

Note: You can upgrade previous versions of the desktop Liferay Sync client to version 3.0. When doing so, however, you need to set up your account again in the new version of the client. Prior to upgrading, it's typically best to shut down Liferay Sync, backup files from your local Sync folder, and delete that folder.

The Liferay Sync client for Mac is packaged in a DMG file. Double-clicking on the DMG file mounts it as a disk image and opens a window showing the image's contents. To install Sync, drag the Liferay Sync icon to your Applications folder. Once it's installed, run it from the Applications folder. If you're running Mac OS X 10.9 or lower, you're prompted for your machine's administrator credentials to install the Finder icon/context menu tool. This prompt only appears when installing or upgrading the tool.

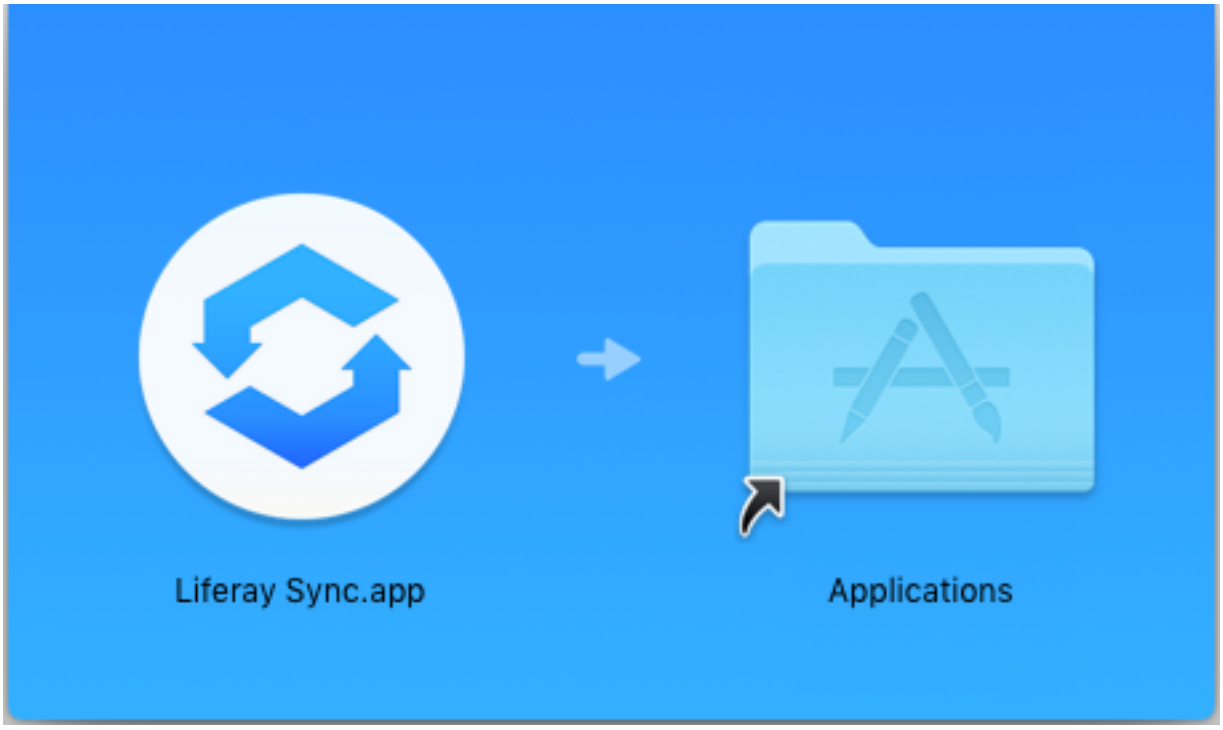


Figure 17.11: Drag the Liferay Sync icon to the Applications folder.

Next, you'll configure the Sync client.

Configuring the Liferay Sync Desktop Client

Now that you've installed Sync, you're ready to configure it! The configuration steps for Sync on Windows and Mac are identical.

1. Open Sync and enter your Liferay DXP instance's address along with your account credentials. Click *Sign In* when you're finished.

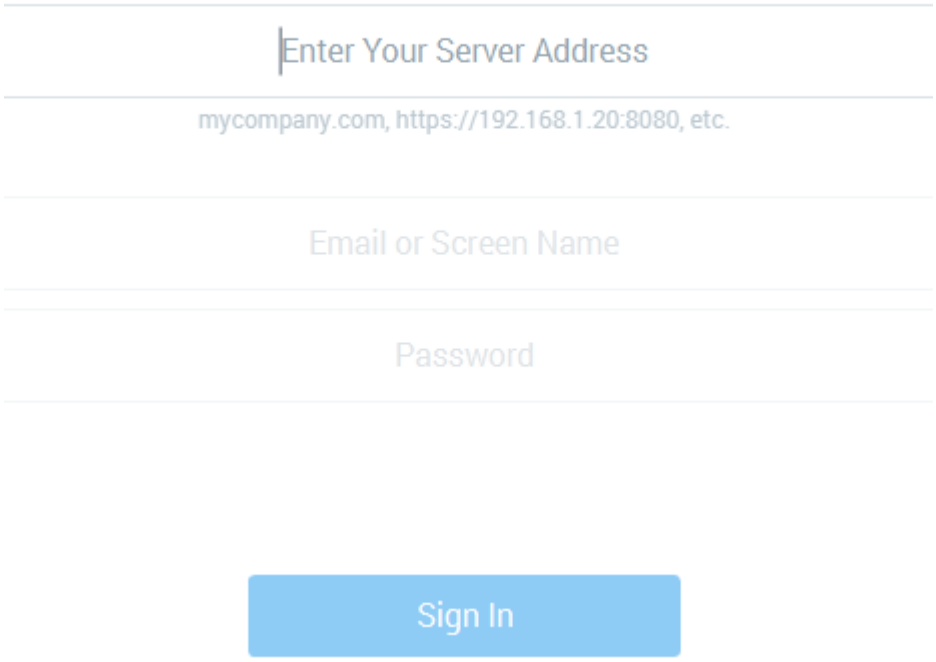


Figure 17.12: The first time you run Liferay Sync, you need to tell it how to communicate with your Liferay DXP server.

When connecting to a server via HTTPS, an error appears if the certificate can't be verified. Choosing *Proceed Anyway* bypasses verification and leaves the connection open to compromise. Liferay Sync attempts to read the certificates specified in the Java Control Panel (see section 20.4.5). If Java isn't installed, you can also put your certificates in `[user.home]/.liferay-sync-3/certificates`. Liferay Sync will trust all certificates in this folder.

2. Select the sites you want to sync with. You can search for a site in the *Search* bar above the site list. If you want to sync all the subfolders of your selected sites, click *Proceed* and move on to the next step.

   To sync only specific folders in a site, first click the site's gear icon. In the window that appears, all folders are selected by default. Unselect the folders you don't want to sync with. Unselecting a subfolder causes the parent folder's checkbox to show a minus sign, indicating that you haven't selected all of the parent folder's subfolders. To sync only the documents at the top of a folder's hierarchy (no subfolders), unselect all of that folder's subfolders. You can also do this by clicking the folder's checkbox until the minus sign appears. Click *Select* when you're finished with your selections, and then click *Proceed* to move on to the next step.

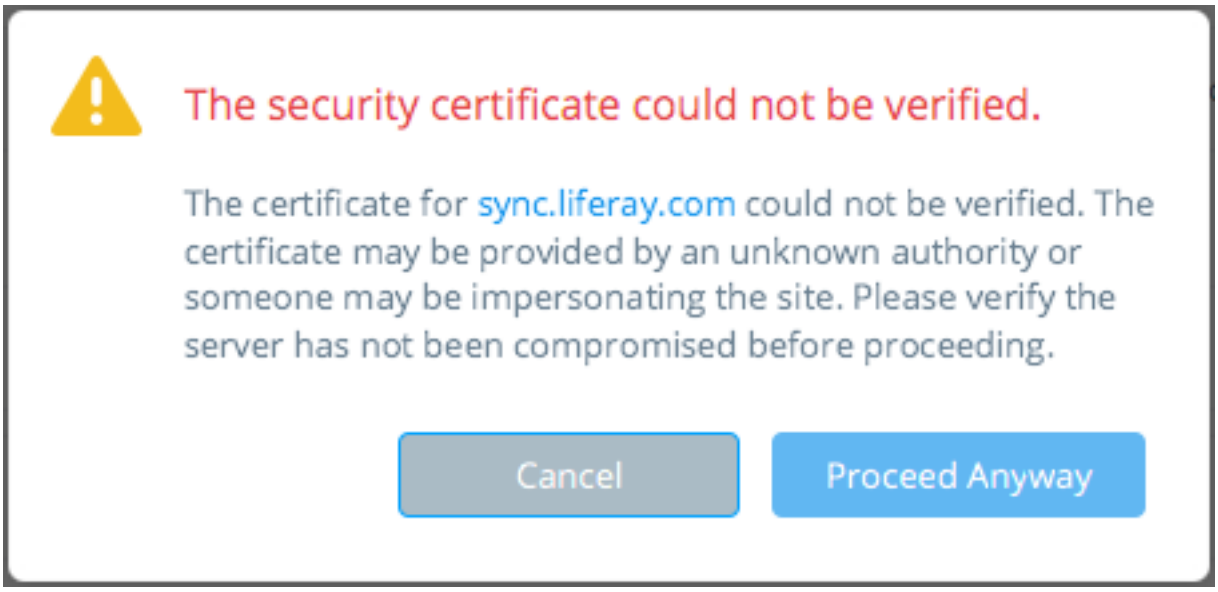


Figure 17.13: When connecting over HTTPS, Liferay Sync produces an error if it can't verify the security certificate. Choosing *Proceed Anyway* bypasses verification and leaves the connection open to compromise.

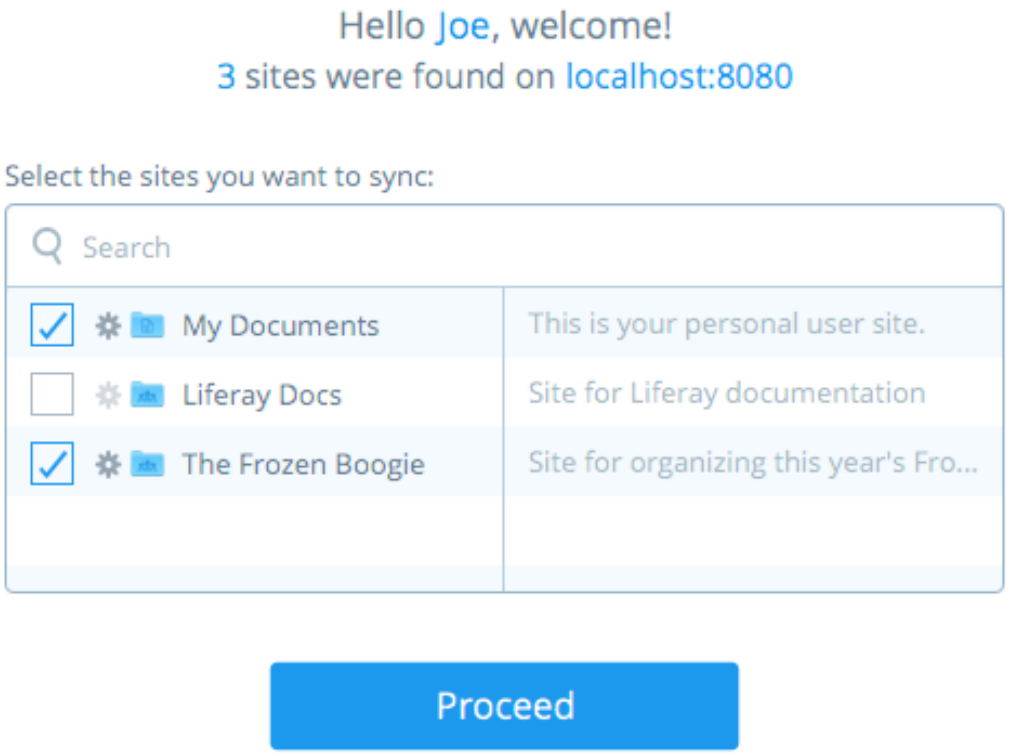


Figure 17.14: Select the sites you want to sync with. Clicking a site's gear icon opens another window where you can choose to sync with only specific subfolders in that site.

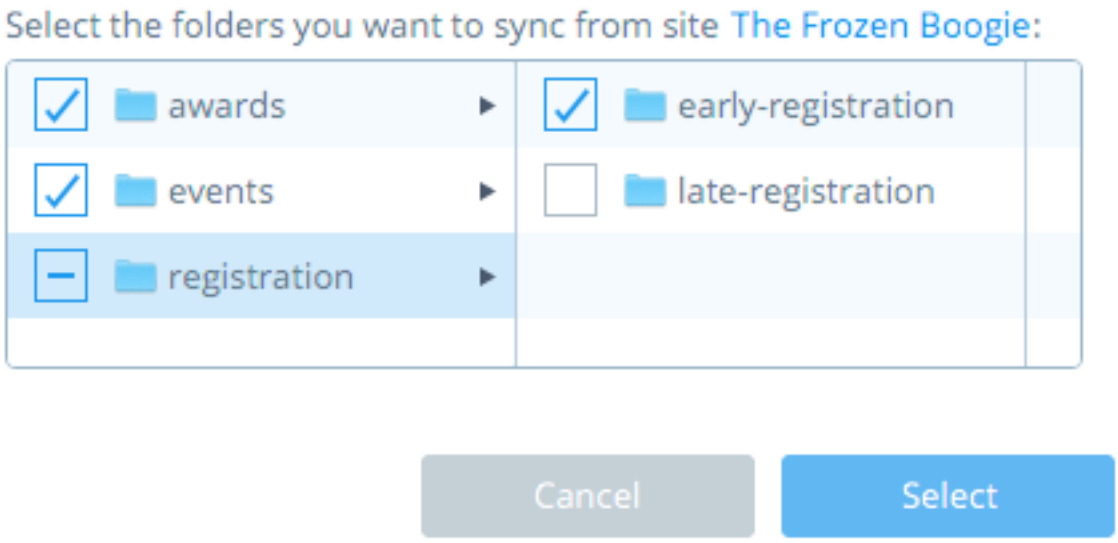


Figure 17.15: Choose the site's subfolders that you want to sync with. The checkbox with the minus sign indicates that not all of the *registration* folder's subfolders are selected.

3. Specify the local folder your Liferay DXP instance will sync with. This folder will be used exclusively for Sync: Sync creates it and it must not conflict with any existing local folder. The Sync folder's default name is the instance's host name, and its default location is the user's documents folder. For example, since the instance in the following screenshots runs locally at the address `http://localhost:8080/`, Sync creates a Sync folder named *localhost* in the user's documents folder. You can, of course, specify any unique name and location for the Sync folder. Click *Start Syncing* to begin syncing files.

```
**Note:** Syncing to network drives is not supported because Liferay Sync
can't reliably detect local file changes on such drives.
```

```
![ Specify your local Sync folder's name and location.](./images/sync-setup-03.png)
```

4. Celebrate! You've successfully set up Liferay Sync! Sync congratulates you on setting it up and begins to sync files from the sites you selected to your local Sync folder. Note, completing the initial synchronization may take a significant amount of time, depending on the amount of data being transferred. You can safely close the window as syncing continues in the background. To view the local Sync folder, click *Open Folder*. To open Sync's preferences, click the small gray text *advanced setup* near the top-right.

Next, you'll learn how to use the Liferay Sync desktop client.

Using the Liferay Sync Desktop Client

When Liferay Sync is running its icon appears in your task bar (Windows) or menu bar (Mac). Clicking this icon opens a menu that lets you work with and manage Liferay Sync.

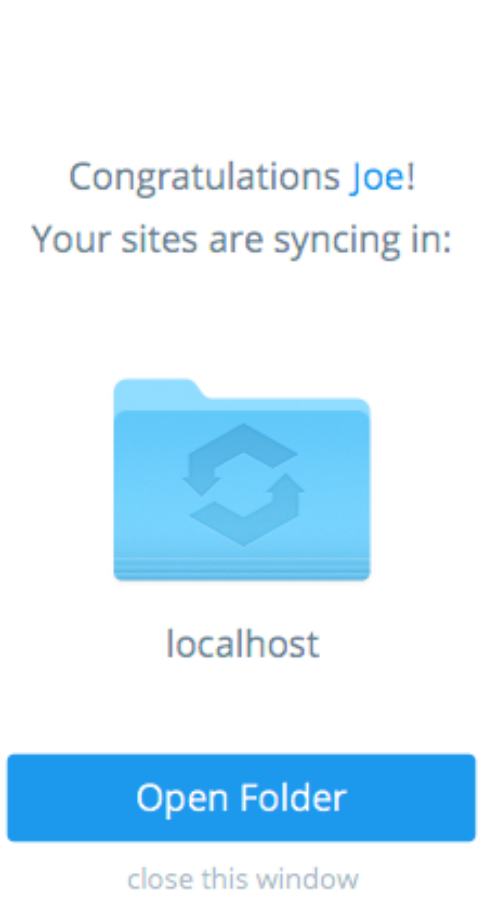


Figure 17.16: Congratulations, you've successfully set up Liferay Sync!

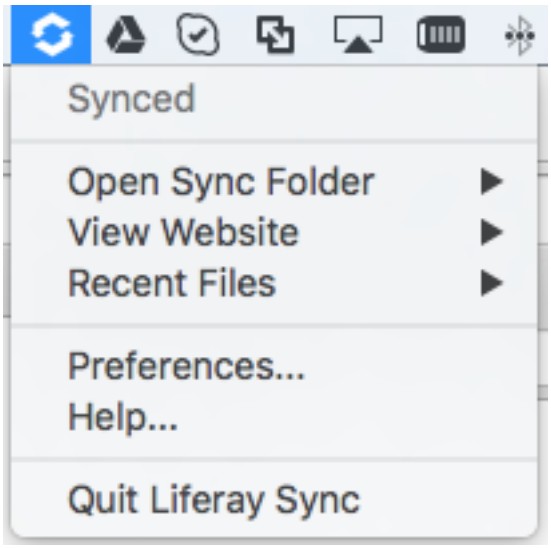


Figure 17.17: The Liferay Sync menu in the Windows task bar and Mac menu bar gives you quick access to Sync.

The top of this menu shows your Sync status. If all your selected sites are synced, then your status is *Synced*.

Below your Sync status, the menu lists three shortcuts for accessing your Liferay DXP instance's files:

- **Open Sync Folder:** Select a site to open its local Sync folder.
- **View Website:** Select a site to view the page in Liferay DXP that contains its Documents and Media app.
- **Recent Files:** Lists recently created and modified files in the repositories you can access.

Note that if you sync with two or more Liferay DXP instances, Sync shows each at the top of the menu instead of your Sync status. Mouse over each instance to reveal a submenu with that instance's Sync status and file shortcuts.

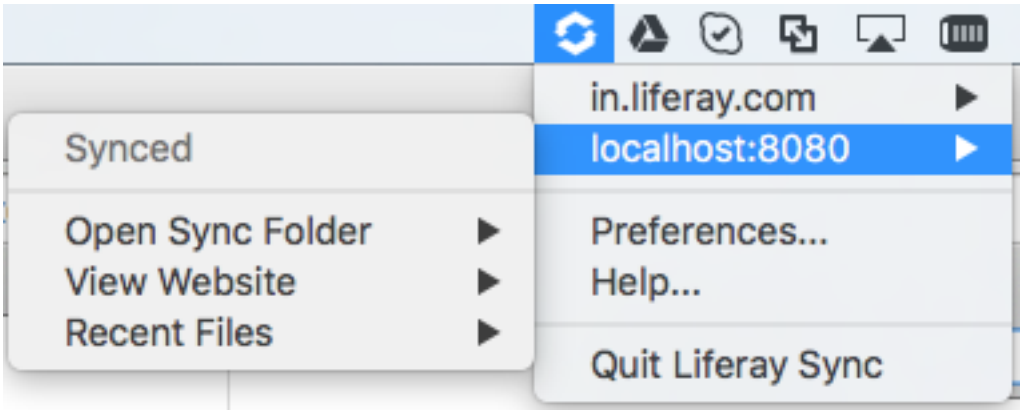


Figure 17.18: When you sync with more than one Liferay DXP instance, Sync shows submenus for each.

Lastly, regardless of how many Liferay DXP instances you sync with, the menu lists the following three options:

- **Preferences:** Open Sync's preferences.
- **Help:** Open Sync's documentation.
- **Quit:** Shut down Sync on your machine.

Next, you'll learn how to use Sync's preferences to control how Sync functions on your machine.

Using Sync Preferences You can use Sync's preferences to add/remove Liferay DXP instances to sync with, edit connection settings, and control Sync's basic behavior. Open Sync's preferences by clicking the Sync icon in the task bar (Windows) or menu bar (Mac OS) and selecting *Preferences*. A preference screen for your instance accounts displays. This is the *Accounts* tab in *Preferences*.

The *Accounts* tab contains the following:

- **Accounts:** the instance accounts you sync with. When you select an account, the sites you have permission to sync with are shown on the right under *Syncing Sites*. You can use the plus, minus, and pencil icons at the bottom of the account list to add, delete, or edit an account, respectively. You should use caution when deleting an account from your Sync client, as doing so also deletes any local files and folders for that account. Adding an account takes you through the same set of steps you used to set up the Sync client. Click here for instructions on this.

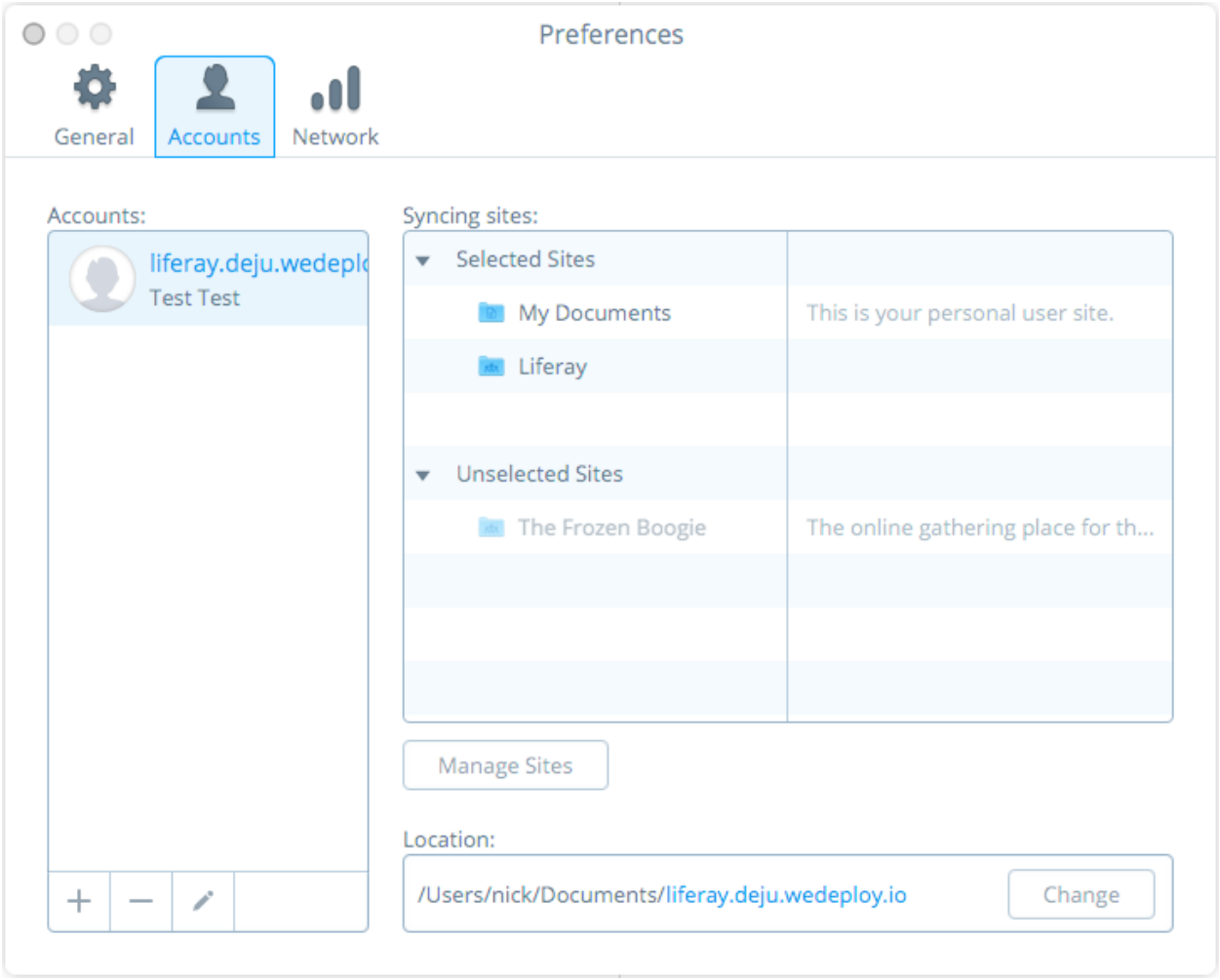


Figure 17.19: : The Preferences menu's *Accounts* tab lets you manage syncing with sites per account.

- **Syncing Sites:** the sites you have permission to sync with for the selected account. The sites you currently sync with are shown under *Selected Sites*. Other sites available for syncing are shown under *Unselected Sites*. To change the sites you sync with, click the *Manage Sites* button. The window that appears lets you select and/or unselect sites to sync with. This window is identical to the one that appeared when you first configured the client. Click here and see step two for instructions on using it. Use caution when de-selecting sites. De-selecting a site deletes its folder on your machine.
- **Location:** the selected account's local Sync folder location. Click the *Change* button to change this folder's location.

The Preferences menu's *General* tab contains settings for the Sync client's general behavior. It lists the following options:

- **Launch Liferay Sync on startup:** starts Sync automatically each time your machine starts.
- **Show desktop notifications:** shows a small notification in the corner of your screen when a synced file changes.

- **Automatically check for updates:** automatically check for new client versions. You can click the *Check Now* button to check for updates manually.

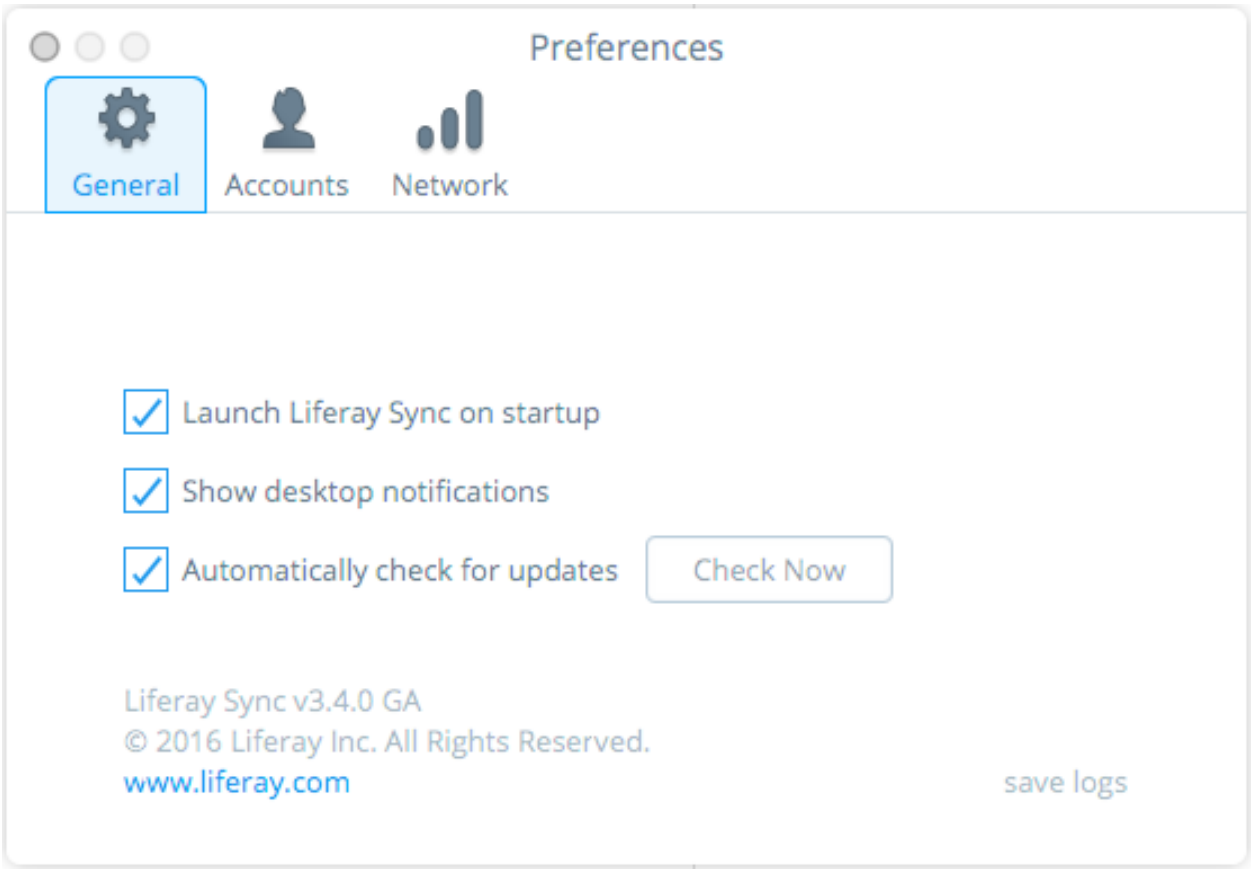


Figure 17.20: : The Preferences menu's *General* tab contains settings for Sync's general behavior.

Lastly, the Preferences menu's *Network* tab lets you control how Sync transfers data with your Liferay DXP instances. It contains the following options:

- **Download Rate:** To limit the rate at which Sync downloads data, select *Limit to* and then specify the rate.
- **Upload Rate:** To limit the rate at which Sync uploads data, select *Limit to* and then specify the rate.
- **Enable LAN Syncing:** Whether to download updates from other desktop clients on the same local network before downloading from the server. This can help reduce server load and increase data transfer speeds. Note that LAN syncing only works when enabled in the Liferay DXP instance by the administrator, and in other clients.

Note that your Liferay DXP instance's administrator can also limit the download/upload rate from the Liferay DXP instance. In this case, the Liferay DXP instance's settings take precedent. For example, if you set a 5.0 MB/s download rate in the client but the Liferay DXP instance's download limit is 2.0 MB/s, the latter takes precedence. Also, the client's rate applies across all its accounts. For example, if the client connects to

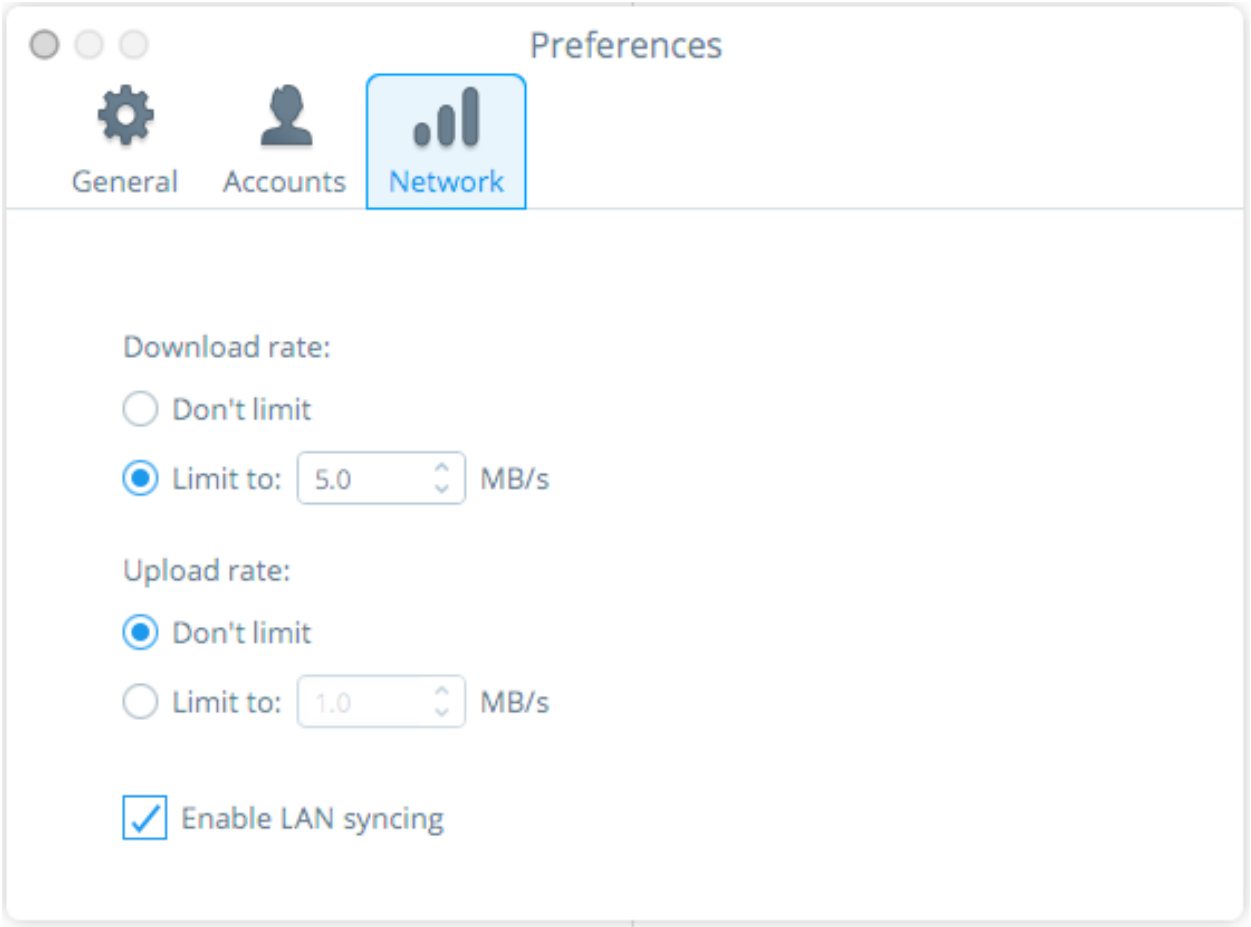


Figure 17.21: : The Preferences menu's *Network* tab contains settings for Sync's data transfer behavior.

three accounts and its download rate is 5.0 MB/s, then the sum of the download rate for all three accounts never exceeds 5.0 MB/s.

Now that you have a handle on configuring Sync, it's time to see how it works with the files in your Sync folder.

Using Your Local Sync Folder Once you configure and run Sync, Sync automatically uploads to your Liferay DXP instance any files you add or modify in your Sync folder. Sync also downloads to your Sync folder any file changes by other users. If you delete a file in your Sync folder, Sync also deletes it from the instance and other clients. You should therefore use **extreme caution** when deleting files in your Sync folder. If you accidentally delete a file though, not all is lost! The file can still be recovered from the instance's Recycle Bin, which is enabled by default. Note, if the instance or site administrator has disabled the Recycle Bin, recovering deleted files is impossible.

Warning: Deleting a file in your Sync folder also deletes it in the Liferay DXP instance and in other clients. If you accidentally delete a file, it can be recovered from the instance's Recycle Bin. The Recycle Bin is enabled by default. File recovery is, however, impossible if the instance or site administrator has disabled the Recycle Bin.

You can run through the following exercise to familiarize yourself with how to create, edit, download, and upload files with Sync. First, open your Sync folder in your file manager and create a new file called `README.txt`. Enter the word `test` in this file. Next, make sure you can access this file in your Liferay DXP site. Go to the site you want to sync with and navigate to its Documents and Media app. It lists your `README.txt` file.

Download the `README.txt` file to a convenient location on your machine. Open the file and check that it still says `test`. Now open the `README.txt` file in your Sync folder and edit it so that it says `second test`. Once the changes are synced, go back to your browser and refresh the page with your Documents and Media app. Click on the *README.txt* file's name, look at the file information displayed, and check that the file's version number has been incremented.

If you download and open the `README.txt` file again, it now says `second test`. Your edit was uploaded to the site! You can be confident that this edit was also downloaded by all other Sync clients connected to your site.

Now delete the `README.txt` file from your local Sync folder. When the changes finish syncing, go back to your browser and refresh the page containing your Documents and Media app. The file is gone! The file is also deleted from the local Sync folders of all other Sync clients connected to the site. Remember this very important rule: deleting files in your local Sync folder deletes them *everywhere*! Next, you'll learn how to use the Sync client for your mobile device.

17.7 Using Liferay Sync on Your Mobile Device

Liferay Sync for Android and iOS contains most of the desktop Sync client's functionality. The mobile client can, however, only be connected to one Liferay DXP instance account at a time. Also, mobile Sync doesn't automatically download files to your device. To save precious storage space on your device, the Sync mobile app lets you choose the files you want to work with. As with the Sync desktop clients, the latest versions of Sync on Android and iOS provide a consistent user experience across platforms. While this article details using Sync on Android, the instructions also apply to Sync on iOS.

You need to download and install Sync on your Android or iOS device through its respective app store, the same as you do any other mobile app. To find the app, search Google Play or the App Store for *Liferay*. You can also download Sync from the Liferay Sync downloads page. Once you've installed the Sync app on your device, follow the instructions below to learn how to use it.

Connecting to Your Liferay DXP Instance

When Liferay Sync first starts on your mobile device, press the *Get Started* button to begin setup. The setup screen asks for your login credentials and your Liferay DXP instance's address. Once you enter them, press *Sign In*. After signing in, the app takes you to a panel that shows your name, a gear icon for accessing the app's settings, and navigation options *My Sites* and *My Documents*. My Sites and My Documents encompass the sites in your Liferay DXP instance that you can sync with. My Documents is your personal user site, while My Sites shows the other sites with which you can sync. No matter how deep you are in the folder hierarchy of a site, swiping to the right returns you to this panel. If you're in the first level of My Sites or My Documents, pressing the location bar at the top slides the screen slightly to the right to reveal a compact view of the panel. The following screenshots show both views of the panel.

Press the gear icon to access Sync's settings. Settings shows your account information and an option to sign out of your Liferay DXP instance. Settings also lets you toggle *Security Mode*. Security Mode protects files stored on your device by encrypting them. Using Security Mode requires you to set up a passcode to use when accessing the Sync app. Security Mode protects the files on your device and Liferay DXP instance in the event your device is lost or stolen. You should note, however, downloading and opening files in Security

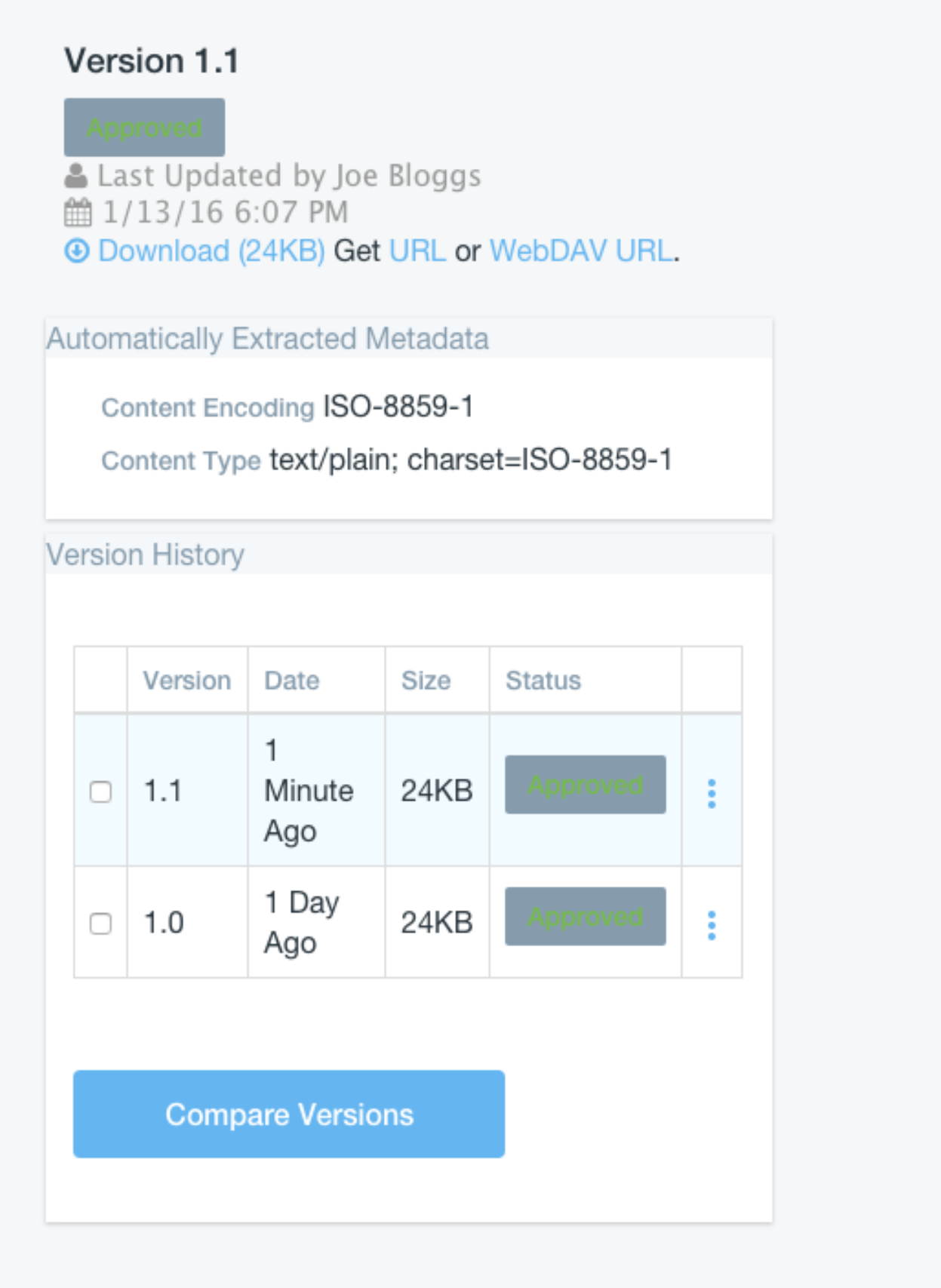


Figure 17.22: : Updating a file through Liferay Sync increments the file's version number. You can view a file's version number through the web interface.

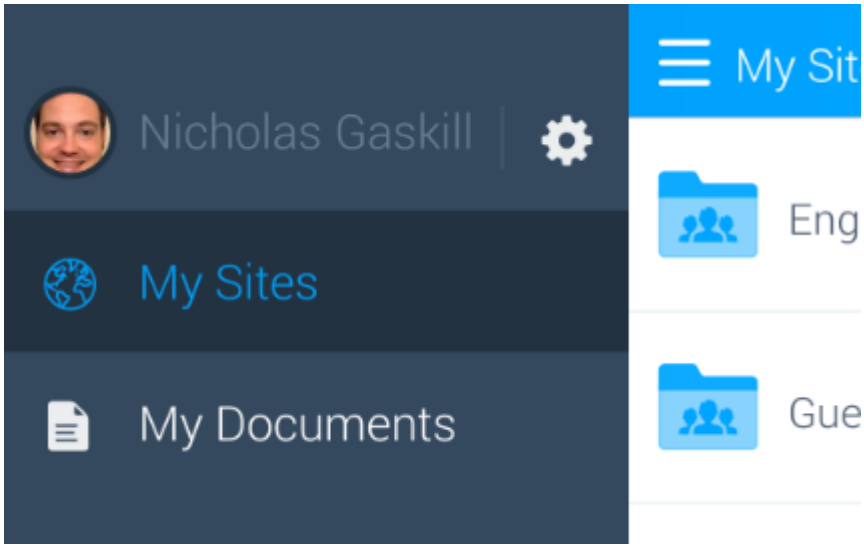


Figure 17.23: This panel lets you access the app's settings, as well as your sites and documents.

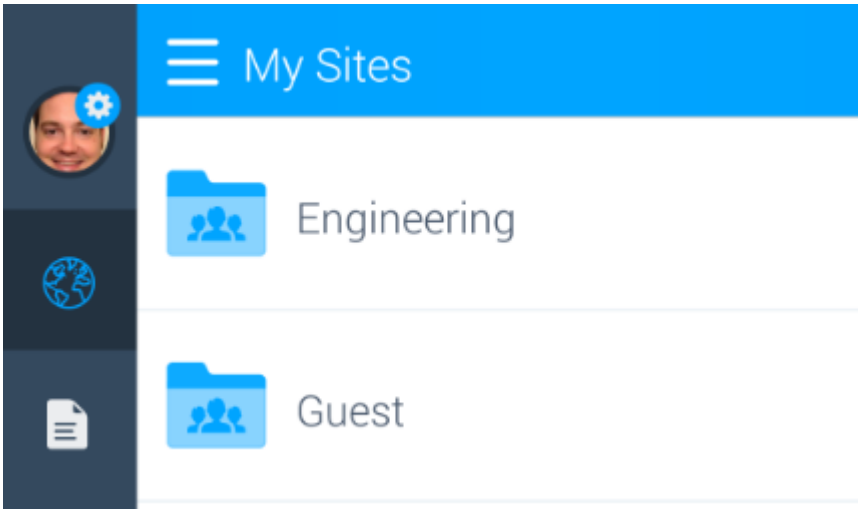


Figure 17.24: Tapping the title bar at the top of My Sites or My Documents opens the main Sync panel's compact view.

Mode takes slightly longer than usual because the Liferay DXP server must use SSL–if it didn't, your files would be transmitting in the open. Below the Security Mode toggle are the app's version and a link to send app feedback to Liferay.

Great! Now that you know how to connect to your Liferay DXP instance and find your sites, you can get started managing the sites' files and folders.

Managing Files and Folders

Whether you're working in My Documents or My Sites, you manage files and folders the same way. Pressing a site or folder shows you a list of its files and folders. It displays each file's size and modification date. You can refresh the list by pulling down from the top of the screen. Your current location in the navigation hierarchy also appears at the top of the screen alongside a plus icon. Pressing the plus icon launches an upload screen that lets you add content in the current location. You can add a new folder, upload a file, or launch your

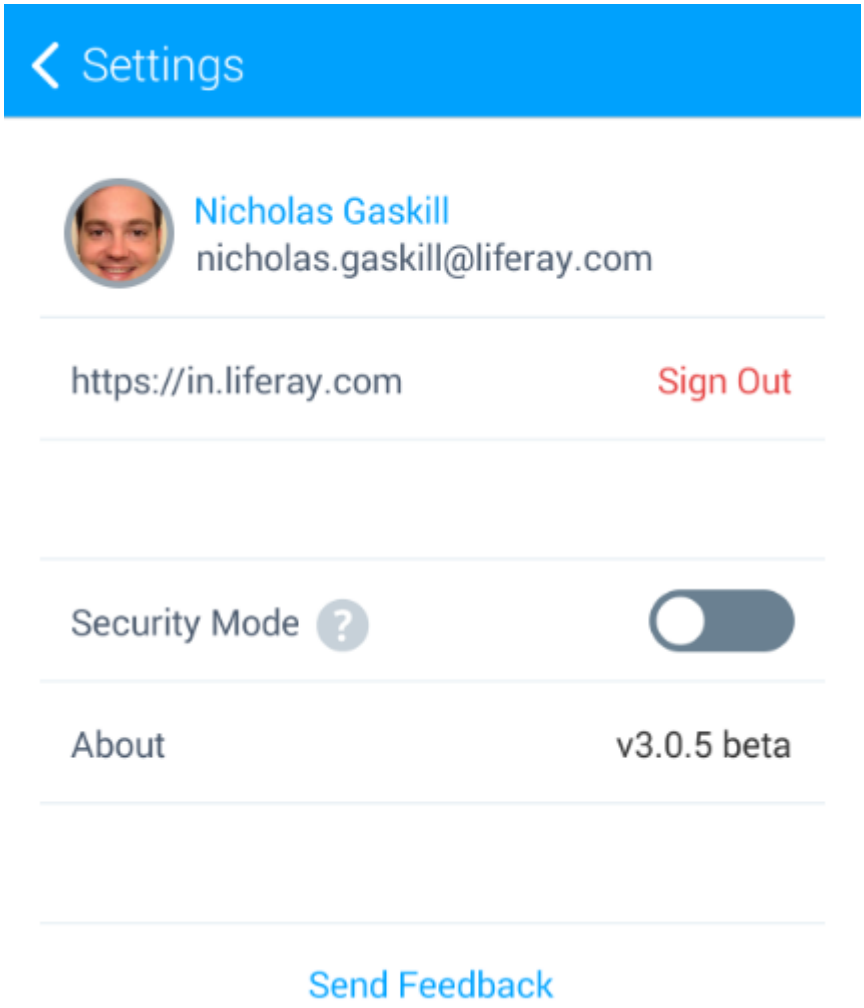


Figure 17.25: The Settings screen for the Sync app lets you sign out of your Liferay DXP instance, enable Security Mode, view the app's version, and send feedback.

device's camera app to take and upload a picture or video. Pressing the *X* icon on the upload screen's top right corner cancels any action and returns you to the current file list.

To download a file to your device, press the file's name in the list. The label that previously showed the file's size and modification date is replaced by a download progress indicator. When the file finishes downloading, your device automatically opens it in the app you've configured to open files of that type. If you haven't configured your device to use a specific app for that file type, you're presented with a list of apps on your device that can open the file. If your device doesn't have an app that can open the file, Sync tells you to install one that can. Downloaded files appear in the list with the file size in blue instead of gray. For example, the screenshot below shows that `LiferayinAction.pdf` is on the device.

The Sync mobile app also lets you move, rename, and delete files and folders. To the right of each file and folder in the list is a circle icon with three dots. Pressing this icon slides open a context menu on the right that lets you move, rename, or delete that item. The screenshots below show these options. Note that you should use **extreme caution** when deleting files or folders. Deleting files or folders in the mobile Sync app also deletes them from your Liferay DXP instance and across any synced clients. Accidentally deleted files can be restored from the instance's Recycle Bin, which is enabled by default. If the instance or site administrator disables the Recycle Bin, however, recovering deleted files is impossible.

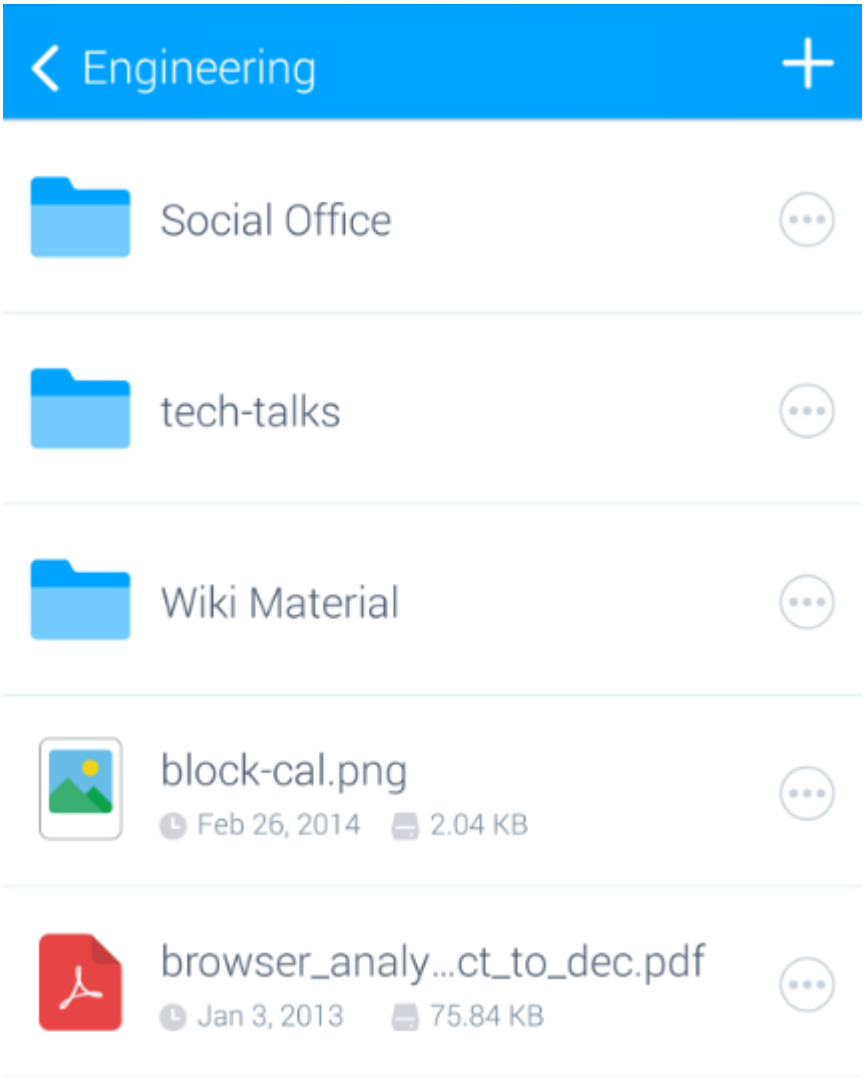


Figure 17.26: Sync shows files and folders in a list.

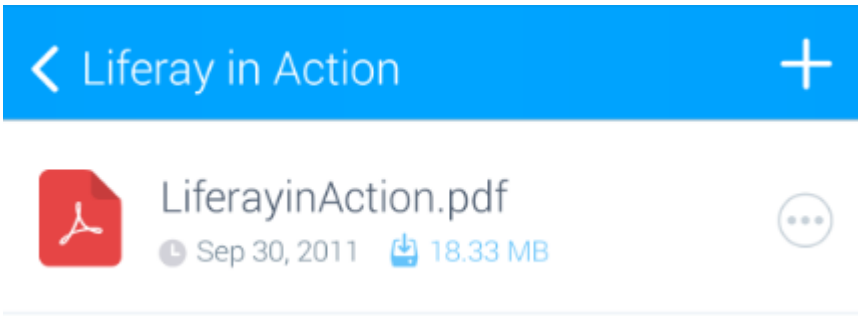


Figure 17.27: Downloaded files appear in the list with their size in blue.

What if you want to delete a file on your device without also deleting it in the instance? Currently, you can only do this by signing out of your account in the app's Settings menu. Doing so removes all downloaded files from your device, but preserves them in the instance. If you're on Android, it may be possible to use a system file browser app to manually remove downloaded files.

Warning: Deleting a file in the mobile Sync app deletes it in the Liferay DXP instance and across any synced clients. If you accidentally delete a file, the instance or site administrator can restore it from the instance's Recycle Bin. The Recycle Bin is enabled by default. If the instance or site administrator disables the Recycle Bin, however, recovering deleted files is impossible.

The context menu also provides additional options for files. A small badge on the file icon's top-right corner indicates the file's version in the Liferay DXP instance. You can also use the context menu to share files you've downloaded. Pressing the *Share* icon opens a list of your device's apps capable of sharing the file. To close the context menu and return to the list of files and folders, swipe to the right. The following screenshot shows the options available in a file's context menu.

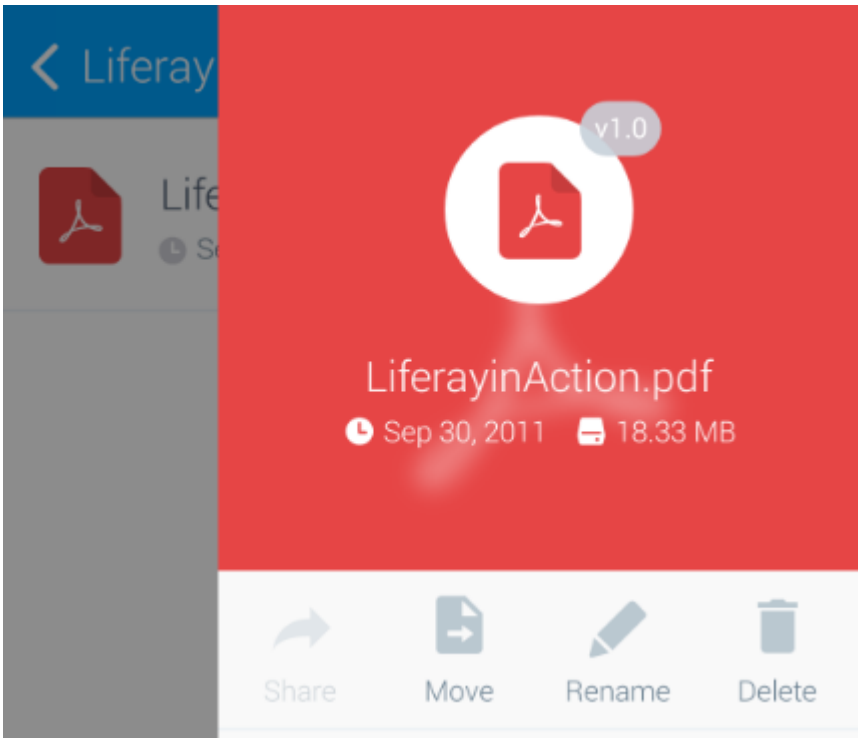


Figure 17.28: The badge on the file's icon shows the file's version in the Liferay DXP instance. You can also share files that you've downloaded.

Awesome! You've seen how Liferay Sync keeps you up to date with the files on your Liferay DXP sites. Any new or modified files on your sites are automatically propagated to your desktop and mobile environments. This lets you view and work with your files from within those environments. What's more, other users you collaborate with can also use Sync to get your latest contributions. This way, entire teams can always work with the latest version of every document.

Chapter 18

Adapting Your Media Across Multiple Devices

Media providers must consider differences between devices (phones, laptops, tablets, etc.) when delivering content: not only their screen size but also their bandwidth and processing capabilities. Liferay DXP's Adaptive Media app allows administrators to control image quality and dynamically adjusts uploaded media to best fit the screen being used.

Note: At this time, Adaptive Media only works for images in blog entries and web content articles.

Adaptive Media integrates with Documents and Media, Blogs, and Web Content. It generates a set of images for use on various screens. When the content is accessed, Adaptive Media checks the screen type and resolution and selects appropriate the appropriate image. You can find the Adaptive Media app on Liferay Marketplace, for Liferay Portal CE 7.0 and Liferay DXP 7.0.

In this section, you'll learn how to manage and use Adaptive Media in Liferay DXP.

18.1 Installing Adaptive Media

You can install the Adaptive Media app from Liferay Marketplace for Liferay Portal CE 7.0 GA5+ or Liferay DXP 7.0. Search for *Adaptive Media* in the Marketplace and follow the app installation instructions at Using the Liferay Marketplace.

Note: To use the Adaptive Media app in Liferay DXP, you must be running Liferay Digital Enterprise 7.0 Fix Pack 35 or higher. See the customer portal to download fix packs.

If you're running Liferay Portal CE 7.0 GA6+, or Liferay DXP 7.0 with Fix Pack 35+, then installing the Adaptive Media app is all you need to do. If you're running Liferay Portal CE 7.0 GA5, however, then you must also install another module. The next section explains how to do this.

Installing Adaptive Media in Liferay Portal CE 7.0 GA5

If you're running Liferay Portal CE 7.0 GA5, then you must install another module after installing the Adaptive Media app:

1. Shut down your Liferay DXP server.
2. Click here to download the module (it's a JAR file).
3. Copy the module into this folder of your Liferay DXP installation:

   ```
   [Liferay_Home]/osgi/marketplace/override/
   ```

   Create the `override` folder if it doesn't exist.
4. Start your Liferay DXP server. You should see these log messages during startup:

   ```
   INFO  [Start Level: Equinox Container: 60d02f03-ed0b-0018-179d-ebc04e65241e][LPKGBundleTrackerCustomizer:411]
       Disabled Liferay CE Web Experience:/com.liferay.journal.service-3.17.3.jar
   INFO  [Start Level: Equinox Container: 60d02f03-ed0b-0018-179d-ebc04e65241e][DefaultLPKGDeployer:393]
       Installed override JAR bundle  LPKG-Override::/liferay-ce-portal-7.0-ga5/osgi/marketplace/override/com.liferay.journal.service.jar
   ```

   These messages indicate that the new module has successfully replaced the old module.
5. You can verify this by running the command `lb "Journal Service"` from the Gogo shell. It should indicate that the module is active:

   ```
   ID|State      |Level|Name
   524|Active     |   10|Liferay Journal Service (3.18.0.ADAPTIVE-MEDIA)
   ```

Well done! Next, you'll learn which modules in the Adaptive Media app are mandatory or optional.

Adaptive Media's Modules

Some modules in the Adaptive Media app are mandatory and must be enabled for Adaptive Media to function, while others can be disabled. The Adaptive Media API modules, which export packages for the other modules to consume, are mandatory; disabling one also disables any other modules that depend on it. Here's a list of the Adaptive Media API modules:

- Liferay Adaptive Media API
- Liferay Adaptive Media Content Transformer API
- Liferay Adaptive Media Image API
- Liferay Adaptive Media Image Item Selector API

The Adaptive Media core modules are also mandatory, and must be enabled to ensure that Adaptive Media works as expected:

- Liferay Adaptive Media Document Library
- Liferay Adaptive Media Document Library Item Selector Web
- Liferay Adaptive Media Document Library Web
- Liferay Adaptive Media Image Content Transformer
- Liferay Adaptive Media Image Implementation
- Liferay Adaptive Media Image Item Selector Implementation
- Liferay Adaptive Media Image JS Web
- Liferay Adaptive Media Image Service
- Liferay Adaptive Media Image Taglib

- Liferay Adaptive Media Image Web
- Liferay Adaptive Media Item Selector Upload Web
- Liferay Adaptive Media Web

The Adaptive Media Blogs modules, which ensure that images uploaded to blog entries can be processed and adapted, are optional. Here's a list of these modules:

- Liferay Adaptive Media Blogs Editor Configuration
- Liferay Adaptive Media Blogs Item Selector Web
- Liferay Adaptive Media Blogs Web
- Liferay Adaptive Media Blogs Web Fragment

The Adaptive Media Journal modules are optional. These modules apply Adaptive Media to web content articles:

- Liferay Adaptive Media Journal Editor Configuration
- Liferay Adaptive Media Journal Web

There are two more optional modules included in Adaptive Media:

- **Liferay Adaptive Media Image Content Transformer Backwards Compatibility:** Ensures that content created before the Adaptive Media installation can use adapted images without the need to edit that content manually. It transforms the images both at startup and when a user views the content, which can negatively affect performance. We therefore recommend that you run some performance tests before using this module in production. You can disable this module if you don't have old content, are experiencing performance problems, or your old content doesn't need adapted images.

- **Liferay Adaptive Media Document Library Thumbnails:** Lets thumbnails in Documents and Media use adapted images. For this to work, you must first migrate the original thumbnails to adapted images. We highly recommend that you enable this module, but it's not mandatory.

Great! Now you know the mandatory and optional modules that come with Adaptive Media. The next section discusses the installation requirements for using animated GIFs with Adaptive Media. If you don't need to use GIFs, you can skip ahead to the article on adding image resolutions to Adaptive Media.

Processing Animated GIFs

To process animated GIFs, Adaptive Media uses an external tool called Gifsicle. This tool ensures that the animation works when the GIF is scaled to different resolutions. You must manually install Gifsicle on the server and ensure that it's on the `PATH`. Once it's installed, you must enable it in Adaptive Media's advanced configuration options.

If Gifsicle isn't installed and `image/gif` is included as a supported MIME type in the advanced configuration options, Adaptive Media scales only a GIF's single frame. This results in a static image in place of the animated GIF.

18.2 Adding Image Resolutions

To use Adaptive Media, you must first define the resolutions for the images delivered to users' devices. Adaptive Media then generates new images scaled to fit those resolutions, while maintaining the original aspect ratio.

To access Adaptive Media settings, open the Control Panel and go to *Configuration* → *Adaptive Media*. Here you can create and manage resolutions.

Note: Adaptive Media configurations apply only to the current Liferay DXP instance.

Once you create a resolution, Adaptive Media automatically generates copies of newly uploaded images in that resolution. Images uploaded before you create the resolution aren't affected and must be adapted separately (see the Generating Missing Image Resolutions section for details).

Adding a New Image Resolution

The number of image resolutions required and the values for each depend on the use case. More resolutions may optimize image delivery, but generating more images requires additional computational resources and storage space. To start, we recommend that you create resolutions to cover common device sizes like mobile phones, tablets, laptops, and desktops. If most users use one device (e.g., all Intranet users have the same company mobile phone), you can create a resolution to target that device.

To add a new resolution, click the *Add* icon ( ) on the Adaptive Media configuration page and provide the following information:

- **Name**: The resolution's name (this must be unique). This can be updated if a custom `Identifier` is defined.
- **Max Width**: The generated image's maximum width. If a *Max Height* is given, this field is optional. This value must be at least `1`.
- **Maximum Height**: The generated image's maximum height. If a *Max Width* is given, this field is optional. This value must be at least `1`.

  Adaptive Media generates images that fit the Max Width and Max Height, while retaining the original aspect ratio. If you only provide one value (either Max Width or Max Height), the generated image scales proportionally to fit within the specified dimension, while maintaining its original aspect ratio. This ensures that adapted images are not distorted.
- **Add a resolution for high density displays (2x):** Defines a scaled up resolution for HIDPI displays. Selecting this option creates a new resolution double the size of the original with the same name and the suffix `-2x`. For example, if the original resolution is `400px` by `300px` (max width by max height), the high density resolution is `800px` by `600px`.
- **Identifier:** The resolution's ID. By default, this is automatically generated from the name. You can specify a custom identifier by selecting the *Custom* option and entering a new *ID*. Third party applications can use this ID to obtain images for the resolution via Adaptive Media's APIs.

Note: Image resolutions and their identifiers can't be updated if the resolution has been used to adapt images. This prevents inconsistencies in generated images.

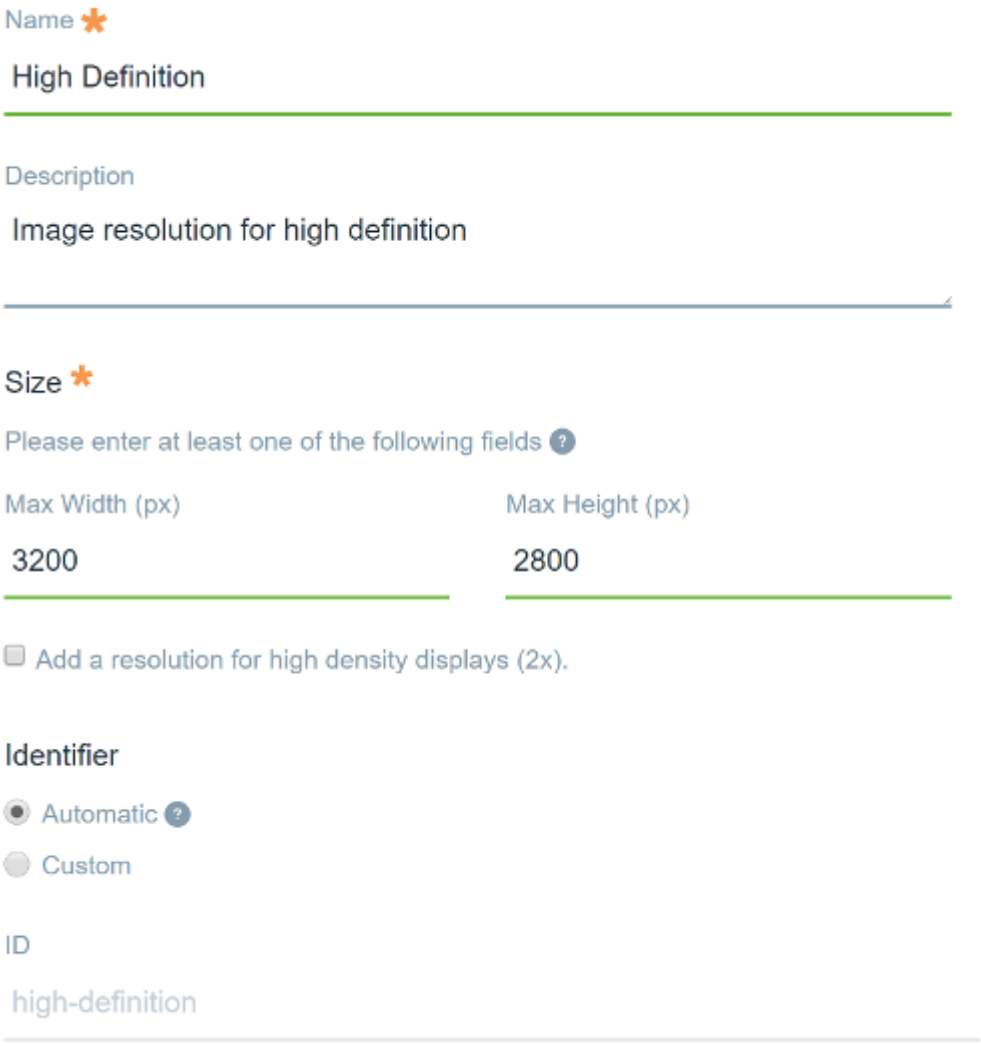


Figure 18.1: The form for adding a new Adaptive Media resolution.

18.3 Managing Image Resolutions

Adaptive Media lets you manage image resolutions and their resulting adapted images. For example, you can disable, enable, edit, and delete resolutions. You can also generate any adapted images that may be missing for a resolution. This article discusses these topics and more.

Disabling Image Resolutions

Disabling an image resolution prevents it from generating adapted images. Any images uploaded after the resolution is disabled use the most appropriate resolution that's still active. Adapted images previously generated by the disabled resolution are still available.

To disable an image resolution, click its *Actions* menu (⋮) and select *Disable*.

Enabling Image Resolutions

Image resolutions are enabled by default. If you need to enable a disabled resolution, click that resolution's *Actions* menu (⋮) and select *Enable*.

While a resolution is disabled, it doesn't generate adapted images for new image uploads. After enabling a resolution, you should generate the adapted images that weren't generated while it was disabled (see the Generating Missing Image Resolutions section for instructions on this).

Editing Image Resolutions

You can't edit an image resolution that already has adapted images. This prevents odd behavior (of the adapted images–you're still free to be as odd as you want). This is because any changes would only be applied to images uploaded after the edit, creating an inconsistent set of adapted images. Odd indeed.

Therefore, editing an image resolution is only possible if Adaptive Media hasn't yet generated adapted images for it. If you must change the values of a resolution that already has adapted images, you must delete that resolution and create a new one with the new values. The next section discusses deleting resolutions.

Deleting Image Resolutions

Be careful when deleting an image resolution, as any adapted images it created are irretrievably lost and are not automatically replaced by new image resolutions you create.

Follow these steps to delete an image resolution:

1. Disable the resolution. You can't delete enabled resolutions. This prevents the accidental deletion of image resolutions.
2. To delete the resolution and all its adapted images, select *Delete* from the resolution's Actions menu (⋮).

Generating Missing Adapted Images

If Adaptive Media hasn't generated all the images you need–say, if new images were uploaded before a new image resolution was created or while the resolution was disabled–you must generate the missing images manually.

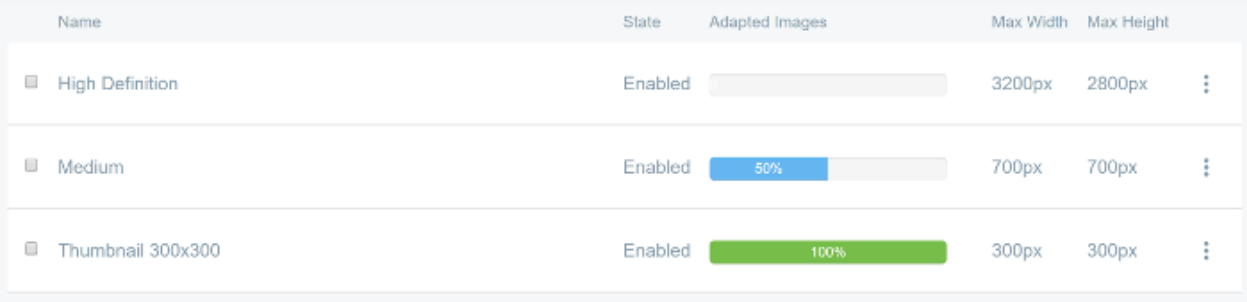


Figure 18.2: The *Adapted Images* column shows the percentage of images that are adapted for each resolution.

To manually generate missing adapted images,

1. For a single resolution, select *Adapt Remaining* from the resolution's Actions menu (⋮).

2. For all resolutions at once, select *Adapt All Images* from the Actions menu in the Control Menu at the top of the page.

The Recycle Bin and Adapted Images

You can't move adapted images directly to the Recycle Bin. But if the original image is in the Recycle Bin, the corresponding adapted images behave as if they are in the Recycle Bin and users can't view them.

Note: URLs that point to adapted images whose original image is in the Recycle Bin return an error code of `404 Not Found`.

If the original image is restored from the Recycle Bin, the adapted images are accessible again.

Awesome! Now you know how to manage image resolutions in Adaptive Media. Next, you'll learn about creating content with adapted images.

18.4 Creating Content with Adapted Images

Adaptive Media is mostly invisible for blog and web content creators. Once an image is added to the content, the app works behind the scenes to deliver an adapted image appropriate to the device in use. Content creators select an image when adding it to their content—they don't have to (and can't) select an adapted image. Adaptive Media identifies each adapted image in the content's HTML with a `data-fileentryid` attribute that is replaced with the latest adapted image when the user views the content. This lets Adaptive Media deliver the latest adapted images to your content, even if the content existed prior to those images.

Note: If Adaptive Media is uninstalled, the original images are displayed in the blog entries and web content articles.

Including Adapted Images in Content

Since Adaptive Media delivers the adapted images behind the scenes, content creators should add images to blog entries and web content as usual: by clicking the image button in the editor and then selecting the image in the file selector.

However, there are some important caveats. When using the file selector to include an image for a blog entry, Adaptive Media works only with images added from the *Blog Images*, *Documents and Media*, and *Upload* tabs. Additionally, adapted images can only be applied to a blog entry's content—cover images excluded. Adaptive Media works for images added to a blog entry via drag and drop, as the image is automatically uploaded to the Blog Images repository, adapted, and then included in the blog entry's content. You can see this by inspecting the HTML and checking that the image contains the `<img>` tag and `data-fileentryid` attribute.

For web content articles, Adaptive Media works only with images added from the file selector's *Documents and Media* tab. **Unlike blogs, Adaptive Media doesn't deliver adapted images for images added to web content articles via drag and drop.**

For both blog entries and media content articles, Adaptive Media doesn't work with images added from the file selector's *URL* tab. This is because the image is linked directly from the URL and therefore provides no image file for Adaptive Media to copy.

Note that you can see the `<img>` tag and `data-fileentryid` attribute in the HTML of a blog entry or a web content article while you're writing it. When the content is displayed, the HTML is automatically replaced and looks similar to this:

```
<picture>

    <source media="(max-width:850px)"
    srcset="/o/adaptive-media/image/44147/med/photo.jpeg">

    <source media="(max-width:1200px) and (min-width:850px)"
    srcset="/o/adaptive-media/image/44147/hd/photo.jpeg">

    <source media="(max-width:2000px) and (min-width:1200px)"
    srcset="/o/adaptive-media/image/44147/ultra-hd/photo.jpeg">

    <img src="/documents/20140/0/photo.jpeg/1992-9143-85d2-f72ec1ff77a0">

</picture>
```

This example uses three different images, each with a different resolution. A `source` tag defines each of these images. Also note the original image (`img`) is used as a fallback in case the adapted images aren't available.

Using Adapted Images in Structured Web Content

To use adapted images in structured web content, content creators must manually include an image field in the web content's structure. Then they can reference that image field in the matching template by selecting it on the left side of the editor. Here's an example snippet of an image field named `Imagecrrf` included in a template:

```
<#if Imagecrrf.getData()?? && Imagecrrf.getData() ≠"">
  <img data-fileentryid="${Imagecrrf.getAttribute("fileEntryId")}"
  alt="${Imagecrrf.getAttribute("alt")}" src="${Imagecrrf.getData()}" />
</#if>
```

This snippet includes the `data-fileentryid` attribute to ensure that Adaptive Media replaces the image with an adapted image. If you inspect the resulting web content's HTML in the editor's code view, you should see a tag like this:

```
<img data-fileentryid="37308"
src="/documents/20143/0/photo.jpeg/85140258-1c9d-89b8-4e45-d79d5e262318?t=1518425" />
```

Note the `<img>` tag with a `data-fileentryid` attribute. Adaptive Media uses the file entry ID to replace the `<img>` element automatically with a `<picture>` element that contains the available adapted images for each resolution (see the `<picture>` example above).

Staging Adapted Images

Adaptive Media is fully integrated with Liferay DXP's content staging and export/import functionality. Adaptive Media includes adapted images in staged content when published, and can update those images to match any new resolutions.

Similarly, when content that contains adapted images is exported, Adaptive Media exports those images in the LAR file. That LAR file can then be imported to restore or transfer that content, along with its adapted images.

Adaptive Media doesn't regenerate adapted images during export/import or the publication of staged content. To improve performance, Adaptive Media instead reuses the existing adapted images.

Awesome! Now you know how create content that contains adapted images. You also know how Adaptive Media includes adapted images in the content's HTML.

18.5 Migrating Documents and Media Thumbnails to Adaptive Media

Liferay DXP automatically generates thumbnails for images in Documents and Media. Once you deploy the Adaptive Media app, however, Liferay DXP doesn't display thumbnails until you migrate them to Adaptive Media. This article walks you through this migration process.

Note: You must be a Portal Administrator to perform the actions described here.

You'll get started by creating image resolutions for the thumbnails in Adaptive Media.

Adding the Replacement Image Resolutions

To migrate the existing Documents and Media thumbnails, you must add new image resolutions in Adaptive Media that have maximum height and maximum width values that match the values specified in the following portal properties:

```
dl.file.entry.thumbnail.max.height

dl.file.entry.thumbnail.max.width

dl.file.entry.thumbnail.custom1.max.height

dl.file.entry.thumbnail.custom1.max.width

dl.file.entry.thumbnail.custom2.max.height

dl.file.entry.thumbnail.custom2.max.width
```

Note: Some of these properties may not be enabled. You need only create image resolutions in Adaptive Media for the enabled properties.

To create the new Image Resolutions, follow the instructions found in the Adding Image Resolutions section of the Adaptive Media user guide.

Now you're ready to to create the Adaptive Media images.

Creating the Adaptive Media Images

Once the required image resolutions exist, you can convert the Documents and Media thumbnails to Adaptive Media images. As mentioned in the Adaptive Media installation guide, the module *Liferay Adaptive Media Document Library Thumbnails* (which is included in the Adaptive Media app) enables this functionality.

There are two different ways to migrate the Documents and Media thumbnails to Adaptive Media:

- **Adapt the images for the thumbnail image resolution:** This scales the existing thumbnails to the values in the Adaptive Media image resolutions, which can take time depending on the number of images. We only recommend this approach when there isn't a large number of thumbnails to process, or if you prefer to generate your images from scratch. This approach is covered in more detail in the Generating Missing Adapted Images section of the Adaptive Media user guide.

- **Execute a migrate process that reuses the existing thumbnails:** This copies the existing thumbnails to Adaptive Media, which performs better because it avoids the computationally expensive scaling operation. The next section describes the steps to run this process.

Running the Migration Process The migration process is a set of Gogo console commands. You can learn more about using the Gogo console in the Felix Gogo Shell tutorial.

Follow these steps to migrate your thumbnails from the Gogo console:

1. Run the `thumbnails:check` command. For each instance, this lists how many thumbnails are pending migration.
2. Run the `thumbnails:migrate` command. This executes the migration process, which may take a while to finish depending on the number of images.
3. Run the `thumbnails:cleanUp` command. This deletes all the original Documents and Media thumbnails and updates the count returned by `thumbnails:check`. Therefore, you should **only** run `thumbnails:cleanUp` after running the migrate command and ensuring that the migration ran successfully and no images are pending migration.

Note: If you undeploy Adaptive Media at some point after running the migration process, you must regenerate the Documents and Media thumbnails. To do this, navigate to *Control Panel → Configuration → Server Administration* and click *Execute* next to *Reset preview and thumbnail files for Documents and Media*.

Great! Now you know how to migrate your Documents and Media thumbnails to adapted images.

18.6 Advanced Configuration Options

Adaptive Media's advanced configuration options are available in System Settings. Open the Control Panel and go to *Configuration → System Settings*, then select the *Collaboration* tab. There are two configurations for Adaptive Media: *Adaptive Media* and *Adaptive Media Image*.

The *Adaptive Media* configuration affects all virtual instances and is related to Adaptive Media's asynchronous processing. These values can be modified to improve performance for specific scenarios or use cases. The following configuration options are available:

- **Max Processes:** The maximum number of processes for generating adapted media. The default value is 5.
- **Core Processes:** The number of processes always available for generating adapted media. The default value is 2. This setting can't exceed the *Max processes* setting.

Warning: Larger values for Max Processes and Core Processes may cause out of memory errors, as processing more images at once can consume large amounts of memory. Out of memory errors can also occur if the source images Adaptive Media uses to generate adapted images are large. You can restrict the maximum size of such images via the *Max Image Size* setting in the *Adaptive Media Image* configuration, which is described next. You should run performance tests to optimize these settings for the amount of memory available on your system.

The *Adaptive Media Image* configuration can be different for each virtual instance. The following configuration options are available:

- **Supported MIME Types:** A list of the image MIME types that Adaptive Media supports. If an image is uploaded and its MIME type isn't in this list, Adaptive Media ignores the image. By default, this list contains many common MIME types.
- **Gifsicle:** To scale animated GIFs, Adaptive Media uses an external tool called Gifsicle. First install Gifsicle on the server, ensure that it's on the `PATH` environment variable, and then click the box next to *Gifsicle Enabled*. If Gifsicle isn't installed and `image/gif` is included as a supported MIME type, Adaptive Media scales only one frame of the GIF, making a static GIF.
- **Max Image Size:** Maximum size of the source images that Adaptive Media can use to generate adapted images. Adaptive Media will not generate adapted images for source images larger than this setting. The default value is 10 MB. To generate adapted images for all source images regardless of size, set this to -1. Since generating adapted images from large source images requires significant amounts of memory, you can specify a lower *Max Image Size* to avoid out of memory errors.

Chapter 19

Collaborating

Liferay DXP ships with a robust suite of collaboration applications that you can use to build communities of users for your site. These applications provide all the features that you would expect of standalone applications outside a portal setting. The difference with Liferay's collaboration apps, however, is that they all share a common look and feel, security model, and architecture. They inherit the strengths of being part of Liferay's development platform so you can use them in combination with Liferay DXP's user management and content management features to build a well-integrated, feature-rich website.

This section shows Liferay DXP's collaboration apps available on Liferay Marketplace and all the components of Liferay's Collaboration suite. You'll learn how to set up and administer

- Blogs
- Message Boards
- Wikis
- Announcements
- Mail
- Knowledge Bases
- Bookmarks

Let's jump right in and begin by exploring Blogs.

19.1 Publishing Blogs

Since blogs are a mainstay on the web, bloggers and blog administrators expect feature-rich environments for publishing blogs. They want both a powerful editor and a pleasurable authoring experience. Blog admins demand an intuitive environment that facilitates configuring blog instances and managing blog entries efficiently. Liferay DXP's Blogs meets these demands.

Its editor delivers features you'd expect in an elegant in-context manner. It has a complete set of WYSIWYG controls that appear where and when you need them. You can stay in this easy-to-use mode or switch to source mode to edit your content's HTML code. In source mode, you can work with light text on a dark background or dark text on a light background. To cap things off, you can open the dual screen HTML editor to see your code rendered in real time. You'll have a satisfying experience creating awesome blog posts.

Lastly, Blogs empowers you to show off your blogs using powerful display apps. You can leverage Liferay DXP's built-in display templates or create your own, to present blogs the way you like them. And you can

now add a beautiful cover image to each of your blog entries, like album covers for your posts. Let's face it–you might not be able to judge a book by its cover, but you're more likely to draw readers to your blog entry by decorating it with an enticing cover image!

By now you're probably chomping at the bit to start blogging, right? Let's get started.

Adding Blog Entries

Each site comes with a built-in blog instance, so let's explore adding an entry to your site's blog. In Site Administration, the Blogs app provides the best place to draft blogs. Click the Menu icon ( ) to open the product menu. Then navigate to *Sites* → *[Site Name]* → *Content* → *Blogs*. The *Entries* screen appears, listing the site's blog entries.

Click the *Add* icon ( ) to create a new blog entry.

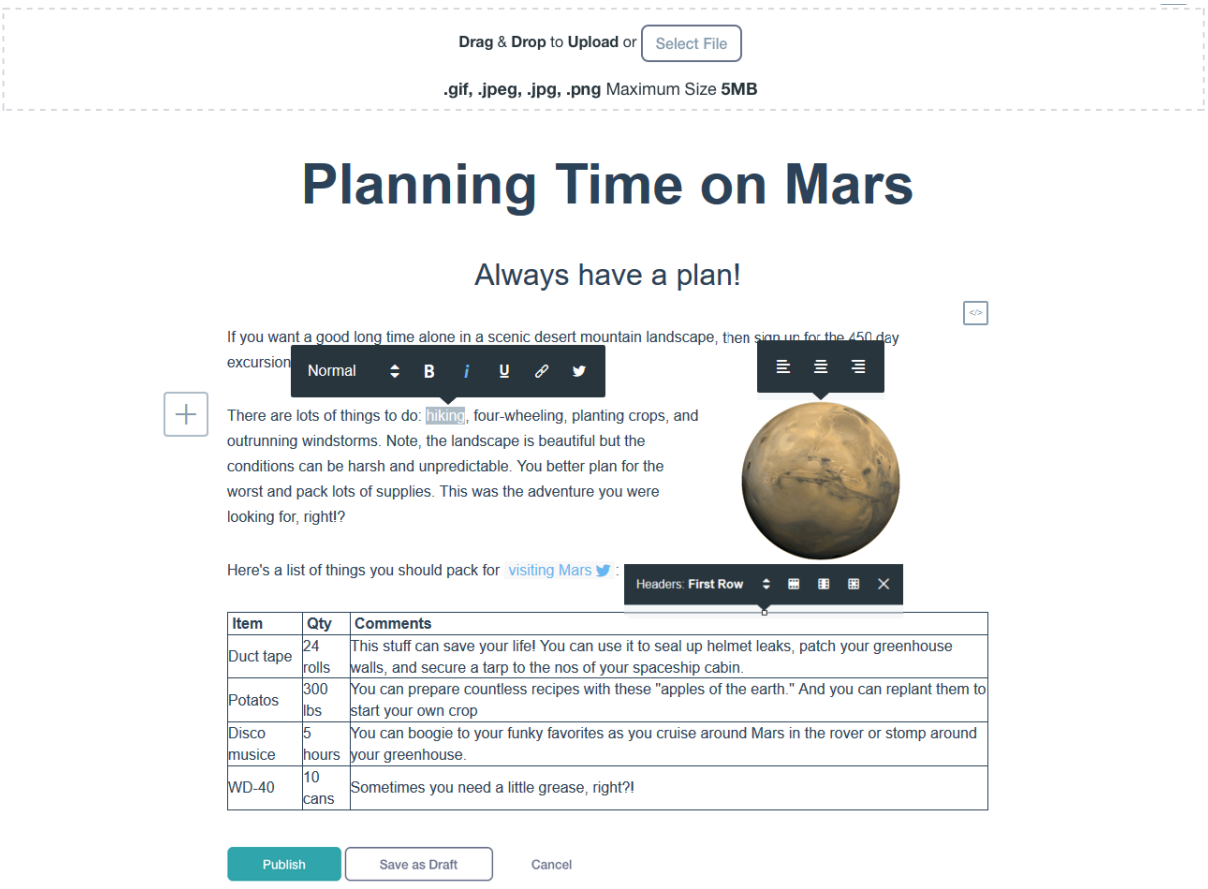


Figure 19.1: This screenshot highlights the blog entry editor's controls for formatting text, justifying images, and editing tables.

The screen presents fields to set a cover image, title, and subtitle, and an area for entering content. You may be asking yourself, "Where are the content editor's controls?" The editor gives you a seamless writing experience, displaying controls when you need them and hiding them from view when you don't need them. This keeps your canvas uncluttered, so you can focus on writing. As you create content, context-specific controls appear.

Go ahead and enter text in the *Content* area. If you highlight text, controls appear (In Figure 1, notice the controls above the highlighted word *hiking*). These controls let you style the text or convert it to a link or a tweet to share your blog post on Twitter. You can also use keyboard shortcuts Ctrl+b for bold, Ctrl+i for italics, and Ctrl+u for underline. You can even mention other users by entering the "@" character followed by the user's name.

Whenever you park your cursor in the content area, the *Add* icon (+) appears. If you click on it, it shows controls for inserting an image, table, or horizontal line ( ). To insert an image, click the icon that depicts the mountain silhouette. The image file selector screen appears. It lets you choose an existing image or upload a new one. You can also drag-and-drop image files into the content area. If you like an image, but want to modify it a bit, use the Image Editor. Select an existing image from the Documents and Media repository and click the pencil icon ( ) in the bottom right corner of the preview window, to open the Image Editor. Any edits you make are automatically applied to a copy of the image.

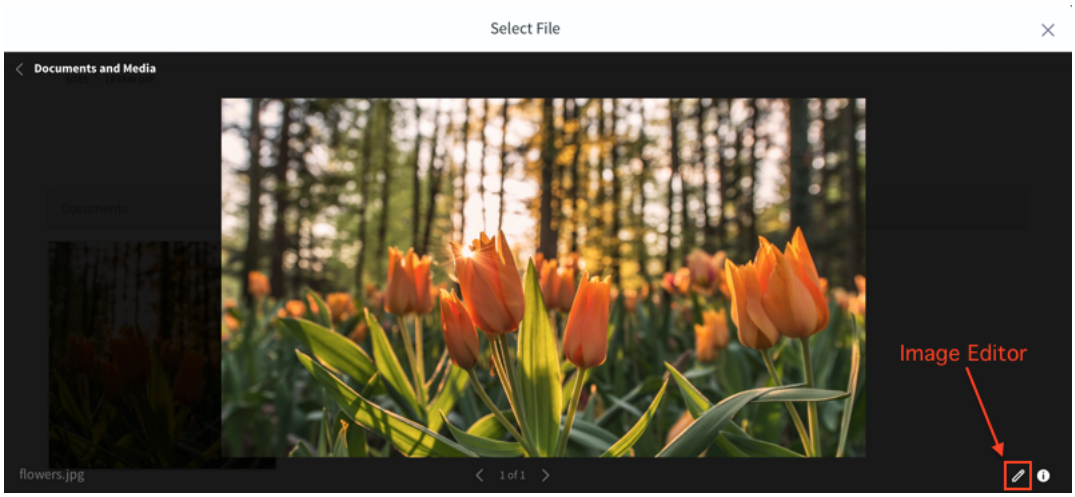


Figure 19.2: You can use the Image Editor to customize your Blog's photos.

After you add an image to the blog entry, clicking the image brings up controls for justifying it to the right or left side of the article. In Figure 1, notice the image justification controls above the moon image.

You can also insert a table with as many rows and columns as you like. If you click inside the table, table editing controls appear (see the table in Figure 1). They let you designate the first row and/or column as table headers. The controls also enable you to add rows, columns, and cells.

Now you're familiar with the editor's regular mode. If you'd rather work with the content's HTML code, you can. To switch the editor to source view, select the *Source* icon (</>). Note that a *Roller* icon ( ) for regular mode appears, giving you the option to switch back to regular view. To satisfy your eyes, source view's moon icon and sun icon let you switch between a dark theme and a light theme.

You can even work in a dual pane view that shows your HTML code on the left and a preview pane on the right. To open this view, click the *Enlarge* icon ( ). You can arrange the HTML and preview panes horizontally or vertically. You can also hide the preview pane, so the HTML editor takes up the entire window space. No matter how you want to use the HTML editor, it can really help you stay "in the zone" as you create awesome blog entries.

Every 25 seconds, the entry you're editing is automatically saved as a draft, so a browser crash or network interruption won't cause you to lose your entry. You can exit the enlarged editor by clicking *Done* (which saves your content) or clicking *Cancel* to abandon any changes since the last auto-save. From the normal-sized source view, you can click the *Roller* icon ( ) to switch back to the WYSIWYG editor.

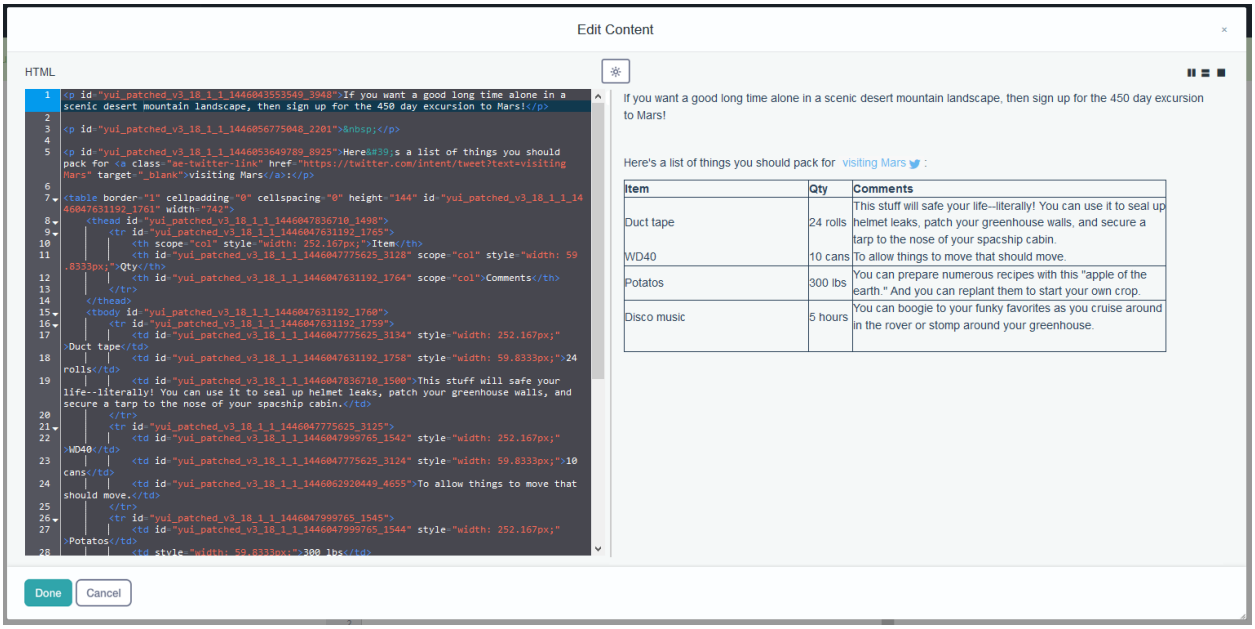


Figure 19.3: The enlarged source editor helps you minimize screen clutter and render changes in real time.

You've learned all the ins and outs of the content editor. Next, you'll learn how to specify your blog entry's other characteristics.

The edit screen's first input field–*Drag & Drop to Upload*–lets you add a cover image (optional) for your entry. This might be an image that represents your entry's purpose and is designed to attract readers. An Asset Publisher shows a blog entry's cover image, by default, as part of the blog entry's abstract. You can drag and drop onto this field any image you like. As an alternative to dragging and dropping an image, you can click the *Select File* button to use the image selector to choose an existing image attached to the blog, an image file from Documents and Media, or an image outside the Liferay DXP instance. You can browse to an image file and upload it. If you select an image from Documents and Media, you have the option to use the Image Editor to customize and edit the image for your cover. Once you've selected the image from Documents and Media, you can access the Image Editor by clicking the pencil icon ( ) in the bottom right corner of the preview window. Any edits you make are automatically applied to a copy of the image, which you can then use as your cover photo.

After you've uploaded the image, it displays in the image pane. To center image content, click on the image and drag it into place with your mouse. Then click on the checkmark icon to save the image placement. You can even add a caption. And if you want to select a different image, you can click the *Change* icon ( ) to use a different image. Lastly, clicking the trash can icon removes the image from the blog entry.

Below the Content area are several panels for configuring your blog entry.

The Categorization panel's options allow you to associate tags and/or categories to your blog entry. Doing this improves search results for blog entries, and you get more navigation options for your users. For example, you can add the Tags Navigation application to another column on your blogs page, which lets users browse blog entries by tag.

Below this is the Related Assets panel. If there's some other content in the Liferay DXP instance that's related to your entry, you can choose it here. For example, you might want to write a blog entry talking about a discussion that happened on the forums. To link those two assets together, select the forum thread under Related Assets.

The Configuration panel is next. Since Liferay DXP 7.0 Fix Pack 13 and Liferay Portal CE GA4, you can specify a friendly URL for the blog entry. The default selection of *Automatic* generates the URL for you based on the blog entry's title. The generated URL appears in the text box shown below the *Blog Entry URL* heading. Alternatively, you can select the *Custom* option to write your own URL. Once published, you can view the blog entry by navigating to the friendly URL. What if you decide to change the blog entry's URL later on? No worries. The original link will redirect to the new friendly URL.

You can also specify an abstract for the blog entry within the Configuration panel. Choose a 400 character text-only abstract or a custom abstract containing a thumbnail image and a manually written description. Below the abstract section, you can set a particular Display Date for the entry.

Lastly for your blog entry's configuration, you can allow *pingbacks*. Pingbacks are XML-RPC requests that are automatically sent when you link to another site. If you link to another site in your blog entry, Liferay DXP sends a pingback to the other site to notify that site that you linked to it. Similarly, if someone links to your blog entry, Liferay can receive a pingback from that person's site and record the link.

The Permissions panel is last. It lets you restrict viewing the blog entry to the owner only or site members (default), or open viewing to site guests. Clicking on the *More Options* link, brings up a permissions table to grant permissions to or revoke permissions from site guests and site members, with regards to the blog entry.

The blog entry permissions enable a role to perform the following actions:

Update: Edit and modify the blog entry.

Permissions: View and modify the blog entry's permissions.

Delete: Move the blog entry to the Recycle Bin.

View: View the blog entry.

Update Discussion: Edit another user's comment on the blog entry.

Delete Discussion: Delete any comments on the blog entry.

Add Discussion: Comment on the blog entry.

Once you've finished your blog entry, click *Publish*. Your blog entry appears with the site's other blog entries.

Congratulations on creating your blog entry! Before displaying it on your site's pages, you may want to learn how to manage blog entries.

Managing Blog Entries

The Blogs application in Site Adminstration helps bloggers and blog administrators manage blog entries. *Search* finds entries that match the keywords you enter. The *Order by* selector enables you to organize entries by *Title* or *Display Date*, in ascending or descending order. Blog entries are displayed using icons, by default. You can select either icon ( ), descriptive ( ), or list ( ) display style. The icon style shows large cover images, descriptive style displays single cell rows that show the author's image and entry information, and list style displays the entries using several columns for each entry. Working with existing blog entries has never been easier!

You can manage entries individually or in a batch. Each entry has an *Actions* icon ( ) to edit the entry, configure its permissions, or move it to the Recycle Bin. You can select the *All* checkbox to select all entries or select an entry's individual check box, and click the trash icon to move them into the Recycle Bin. Under the *Images* tab you can view individual images and delete them individually or in a batch.

The *Options* icon ( ) at the top of Blogs lets you configure permissions and notifications, or import/export the blog. Here are the blog instance Configuration options:

Email From: defines the *From* field in the email messages that users receive from Blogs.

Entry Added Email: defines a subject and body for the emails sent when a new blog entry has been added.

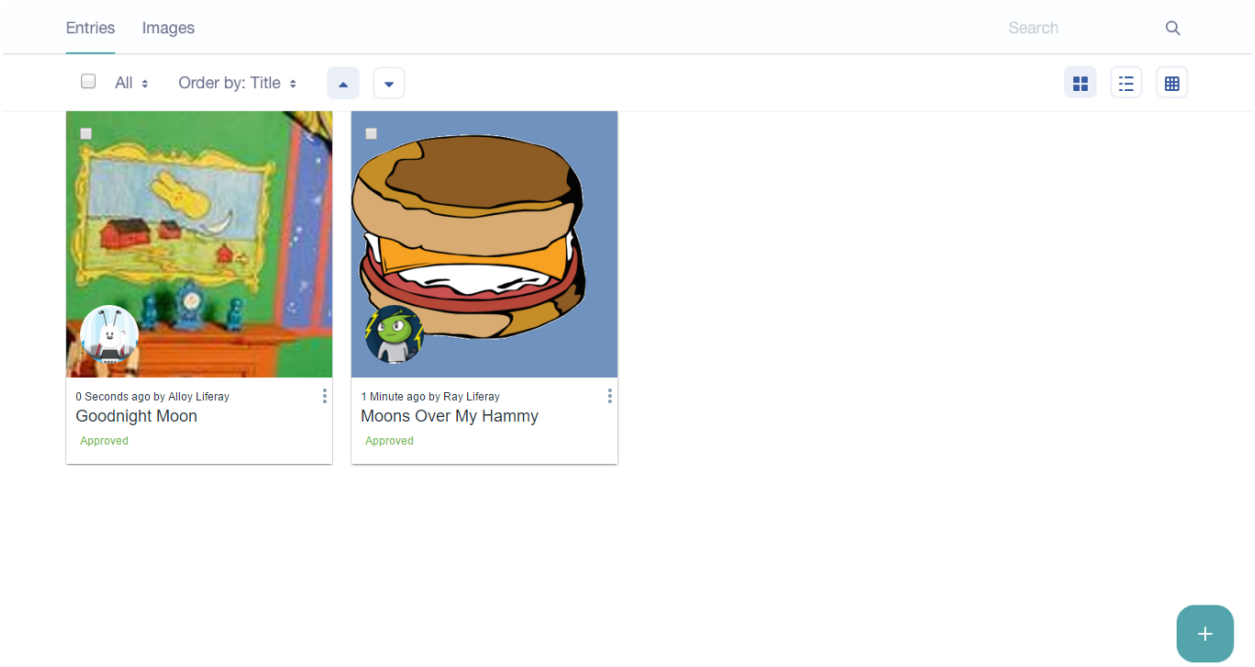


Figure 19.4: Blogs in Site Administration is the perfect place to create and manage blog entries. It has several options for modifying, displaying, filtering, and finding entries.

Entry Updated Email: defines a subject and body for the emails sent when a new blog entry has been updated.

RSS: lets you enable RSS subscription and choose how blogs are displayed to RSS readers.

- **Maximum Items to Display:** choose the total number of RSS feeds entries to display on the initial page. You can choose up to one hundred to be displayed.
- **Display Style:** choose between *Full Content*, *Abstract*, and *Title*. These options work just like the ones above for blog entries.
- **Format:** choose which format you want to deliver your blogs: Atom 1.0, RSS 1.0, or RSS 2.0.

You've learned how to create blog entries and manage them. It's time to consider blog security. For example, who is allowed to view the different blog instances and who is allowed to add blog entries?

If you have a personal blog, the default permissions should work well for you. If you have a shared blog, you may want to adjust its permissions. The default settings make it so only the owner of the site where the application was added can add entries. This, of course, is great if the Blogs app has been added to a user's personal pages, but doesn't work so well for a shared multi-author blog. But don't worry: it's easy to grant blogging permissions to users.

First, create a role for your bloggers and add them to the role. Next, in Blogs, click *Options* → *Entries Permissions*. A list of both instance and site roles is displayed, and currently only the owner is checked for the *Add Entry* permission. Mark any other role or team that should be able to add blog entries and then click *Save*. Once you've done this, users in the roles or teams that you selected are able to post to the shared blog. You can also grant roles and teams the ability to subscribe to blog updates.

From within the Control Panel, you can configure all the permissions for Blogs. Permissions for the Blogs and Blogs Aggregator applications are covered next.

You've now created a blog entry and learned how to manage blog entries and permissions. Next, you'll learn how to use the Blogs and Blogs Aggregator applications to display blog entries the way you want them.

Displaying Blogs

You can add the Blogs application a page from the *Add* menu. Since Blogs supports scopes, you can use it to create a multi-author blog to build a site like http://slashdot.org/ or to create multiple personal blogs to build a site like http://blogger.com. What's the difference? Adding the Blogs app to a site page creates a shared blog for members of the site. Adding the app to a user's personal site (dashboard) creates a blog just for that user. Blogs works the same way in both cases. And of course, you can scope a blog to a page to produce a blog instance for just that page.

Add the Blogs application to one of your site's pages. It lists abstracts of the site's recently published blog entries. Notice that each entry's cover image shows prominently in the listing. The figure below shows a blog entry abstract.

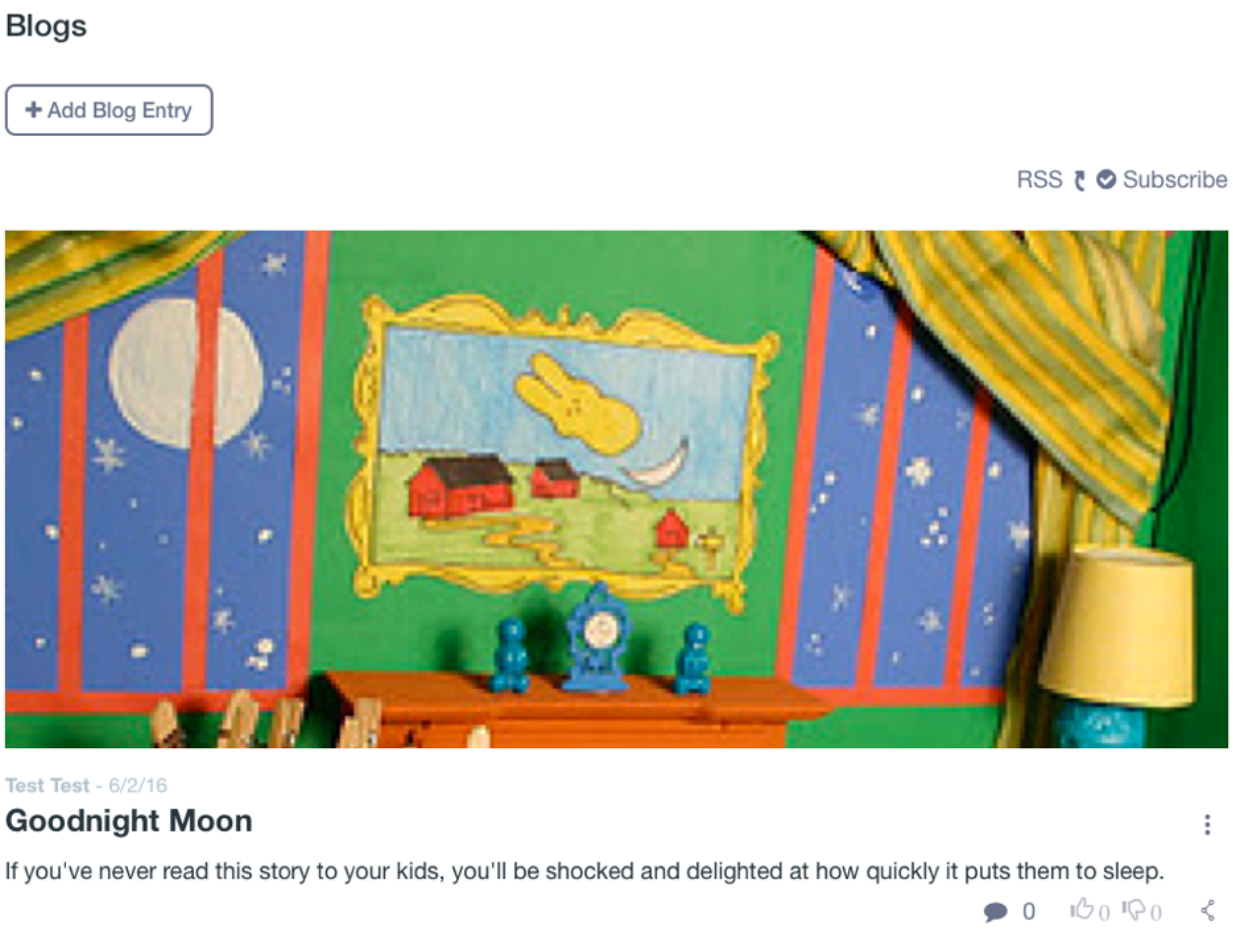


Figure 19.5: Here's a blog entry in abstract format.

You can see that in the summary view, you don't see the trackback/pingback link, and you only see the number of comments that have been added. If you click the entry's title, you'll see the whole article, all the

comments in a threaded view, and the trackback/pingback link which others can use to link back to your blog entry.

The full view of the blog entry also contains links to share blog entries on social networks, such as Twitter, Facebook, and Google+. This gives your readers an easy way to share blog entries with friends, potentially driving further traffic to your site.

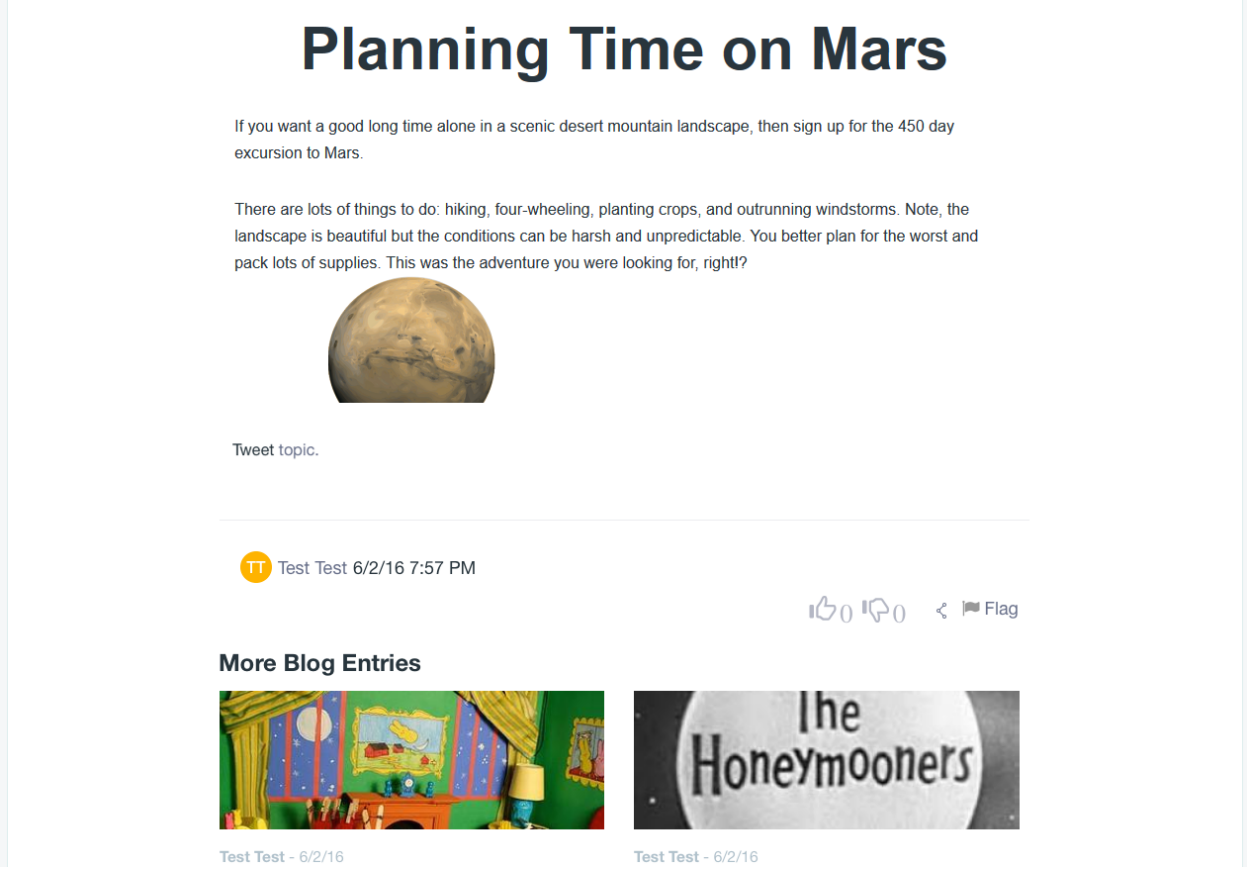


Figure 19.6: Users can view your blog entry in all its glory. They can rate it, comment on it, and share it with other people.

By default, the Blogs application displays the abstract and image of the latest entries. There are several display options that let you configure the listing to look the way you want. To configure the application, click the *Options* icon in the app's title bar and select *Configuration*. The *Display Settings* are in the Setup tab. To choose the right settings, you should think about the best way to display your entries as well as how you want users to interact with bloggers.

Here are the Display Settings:

Maximum Items to Display: choose the total number of blog entries to display on the initial page. You can select up to 75 to display at once.

Display Template: choose between *Abstract*, *Full Content*, *Title*, or *Basic*. Setting this to Abstract shows the entry's cover image and first four hundred characters of text. Users can click on the entry's title to see its full content. You can click *Manage Display Templates for Liferay* to select an existing Blogs application display template (ADT) or to create your own. To learn how to customize your own display templates with Application Display Templates visit the *Using Application Display Templates* section of this guide.

Enable Flags: flag content as inappropriate and send an email to the administrators.

Enable Related Assets: select related content from other applications to pull into their blog entry for readers to view.

Enable Ratings: lets readers rate your blog entries from one to five stars.

Enable Comments: lets readers comment on your blog entries.

Enable Comment Ratings: lets readers rate the comments which are posted to your blog entries.

Enable Social Bookmarks: lets users tweet, Facebook like, or +1 (Google+) blog posts. You can edit which social bookmarks are available in the *Social Bookmarks* section of the Configuration menu.

Display Style: select a menu, simple, vertical, or horizontal display style for your blog posts.

Display Position: choose a top or bottom position for your blog posts.

Social Bookmarks: choose social bookmarks to enable for blog posts, which includes Twitter, Facebook, and Google+.

Here are descriptions for the other Blogs Configuration tabs:

Permissions: shows Liferay DXP's permissions dialog for the Blogs application.

Communication: lists public render parameters the application publishes to other applications on the page. Other applications can read these and take actions based on them. For each shared parameter, you can specify whether to allow communication using the parameter and select which incoming parameter can populate it.

Sharing: lets you embed the application instance as a widget on on any website, Facebook, or Netvibes, or as an OpenSocial Gadget.

Scope: lets you specify the blog instance the application displays: the current site's blog (default), global blog, or page blog. If the page doesn't already have a blog instance, you can select scope option *[Page Name] (Create New)* to create a page-scoped blog instance for the Blogs application to display.

Liferay DXP's Blogs application excels at aggregating information from multiple places. The Blogs Aggregator application lets you "bubble up" blog entries from multiple users and highlight them on your site. Let's look next at how that works.

Aggregating Blogs

You can set up a whole web site devoted just to blogging if you wish. The Blogs Aggregator lets you publish entries from multiple bloggers on one page, giving further visibility to blog entries. You can add it to a page from the *Collaboration* category in the *Add* ( ) → *Applications* menu.

If you click *Configuration* from the *Options* icon ( ) in the app's title bar, the Blogs Aggregator's configuration page appears. From here, you can set its options.

Selection Method: select Users or Scope here. If you select Users, the Blogs Aggregator aggregates the entries of every blogger on your system. If you want to refine the aggregation, you can select an organization by which to filter the users. If you select Scope, the Blogs Aggregator contains only entries of users who are in the current scope. This limits the entries to members of the site where the Blogs Aggregator resides.

Organization: select which organization's blogs you want to aggregate.

Display Style: select from several different styles for displaying blog entries: *Body and Image*, *Body*, *Abstract*, *Abstract without Title*, *Quote*, *Quote without Title*, and *Title*.

Maximum Items to Display: select the maximum number of entries the app displays.

Enable RSS Subscription: creates an RSS feed out of the aggregated entries. This lets users subscribe to an aggregate feed of all your bloggers. Below this checkbox, you can configure how you want the RSS Feed displayed:

- **Maximum Items to Display:** select maximum number of RSS items to display.

- **Display Style:** select from several different styles for displaying RSS feeds: *Abstract*, *Full Content*, and *Title*.

- **Format:** select which web feed language to use for your feed, which includes *Atom 1.0*, *RSS 1.0*, or *RSS 2.0*.

Show Tags: for each entry, displays all the tags associated with the blogs.

At the top of the Configuration screen, there's an option called *Archive/Restore Setup*. It lets you store your current Setup configuration or apply an existing archived Setup. This is especially helpful when you have multiple configurations you like to use in Blogs Aggregator instances.

Here are descriptions for the other Blogs Aggregator's Configuration screens:

Permissions: shows Liferay DXP's permissions dialog.

Communication: lists public render parameters the application publishes to other applications on the page. Other applications can read these and take actions based on them. For each shared parameter, you can specify whether to allow communication using the parameter and select which incoming parameter can populate it.

Sharing: lets you embed the application instance as a widget on on any website, Facebook, or Netvibes, or as an OpenSocial Gadget.

Scope: lets you specify the blog instance the application displays: the current site's blog (default), global blog, or the page's. If the page doesn't already have a blog instance, you can select scope option *[Page Name] (Create New)* to create a page-scoped blog instance and set the Blogs app to display it.

When you've finished setting the options, click *Save*. Then close the dialog box. You'll notice the Blogs Aggregator looks very much like the Blogs application, except that it shows entries from multiple blogs.

Last up is showing your Liferay DXP instance's recent bloggers.

Highlighting Recent Bloggers

A great way to applaud bloggers is to display their names using the Recent Bloggers app. It lists the names of users who've most recently posted blog entries. It shows their name, profile picture, and number of posts. It's a great way to show off who's who on your site. You can add the Recent Bloggers app to a page from the *Collaboration* category in the *Add* ( ) → *Applications* menu.

If you click *Configuration* from the *Options* icon ( ) in the app's title bar, the Recent Blogger's configuration page appears. From here, you can set its options.

Selection Method: select Users or Scope here. If you select Users, the Recent Bloggers aggregates all the users on your system. If you want to refine the aggregation, you can select an organization by which to filter the users. If you select Scope, the app will show only users who are in the current scope. This limits the users shown to the site where the resides resides.

Organization: select which organization's users you want to show.

Display Style: select from several different styles for displaying users: *User Name and Image* and *User*.

Maximum Items to Display: select the maximum number of users the app displays.

Sharing: lets you embed the application instance as a widget on on any website, Facebook, or Netvibes, or as an OpenSocial Gadget.

Scope: lets you specify the blog instance the application displays: the current site's blog (default), global blog, or the page's. If the page doesn't already have a blog instance, you can select scope option *[Page Name] (Create New)* to create a page-scoped blog instance and set the Blogs app to display it.

When you're finished setting the options, click *Save*. Then close the dialog box.

Add the Recent Bloggers app to a page on your site to draw attention to the users who are contributing helpful blogs on your site.

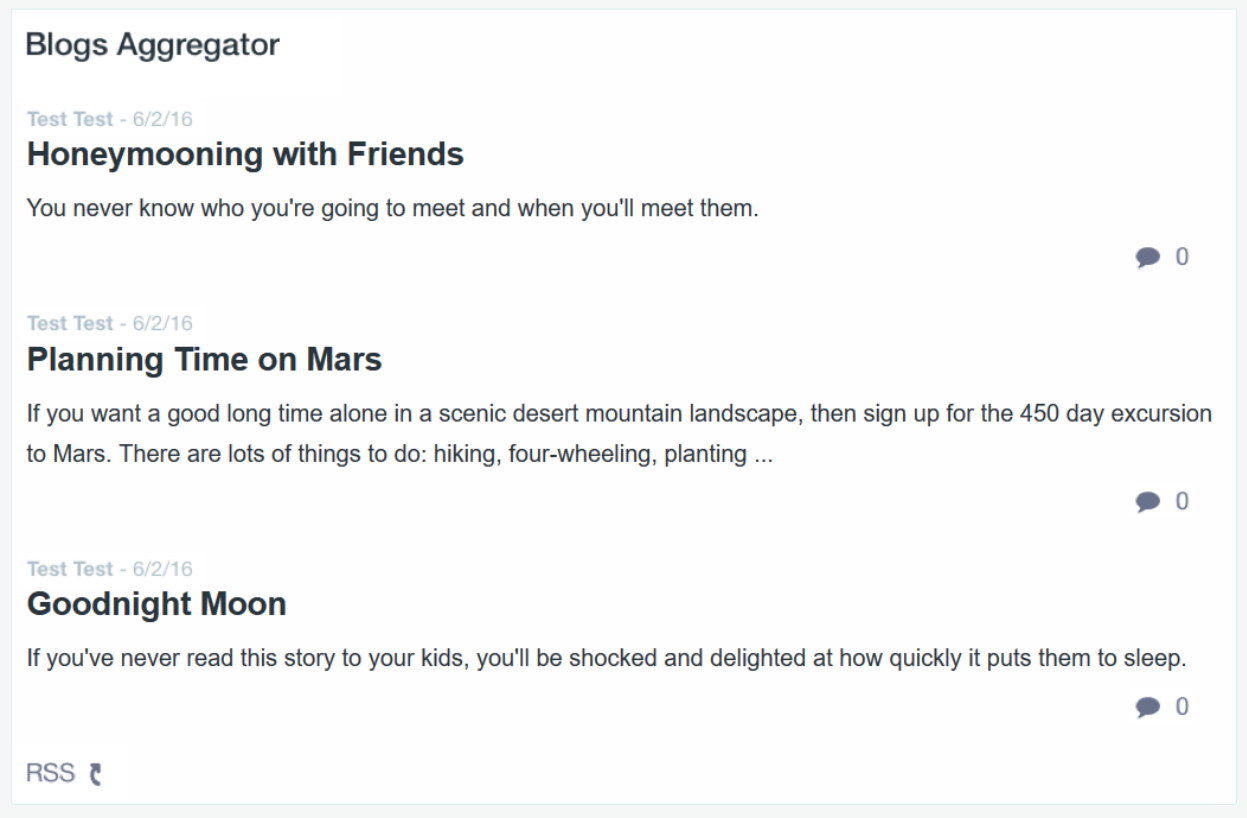


Figure 19.7: The Blogs Aggregator lets you display blog entries authored by multiple authors from different sites.

Summary

Throughout this section, you've learned how to create blog posts in a powerful no-nonsense editor, manage blog instances and blog entries, and display blogs the way you like them using the the Blogs and Blogs Aggregator applications. Now you can enjoy the benefits of participating in the world of blogging.

Blogs are terrific for sharing information on a topic, posting important announcements, and expressing yourself. And if comments are enabled, users can have short exchanges about your blog entry. The best place to discuss things or ask questions, however, is in a forum. Next, you'll learn how to create forums using Message Boards.

19.2 Creating Forums with Message Boards

Liferay DXP's Message Boards app is a state of the art forum application similar to many forums in which you may have participated. The difference, of course, is that Liferay's message boards can inherit the abilities of Liferay DXP to provide an integrated experience that others cannot match.

There are countless web sites out there where it is clearly evident that there is no link whatsoever between the main site and the message boards. In some cases, users are even required to register twice: once for the main site and once for the message boards. Sometimes it is three times: for the site, for the message boards, and for the shopping cart. By providing a message boards application along with all of the other applications, Liferay provides a unique, integrated approach to building sites. You can concentrate on building your site while Liferay DXP does the integration work for you.

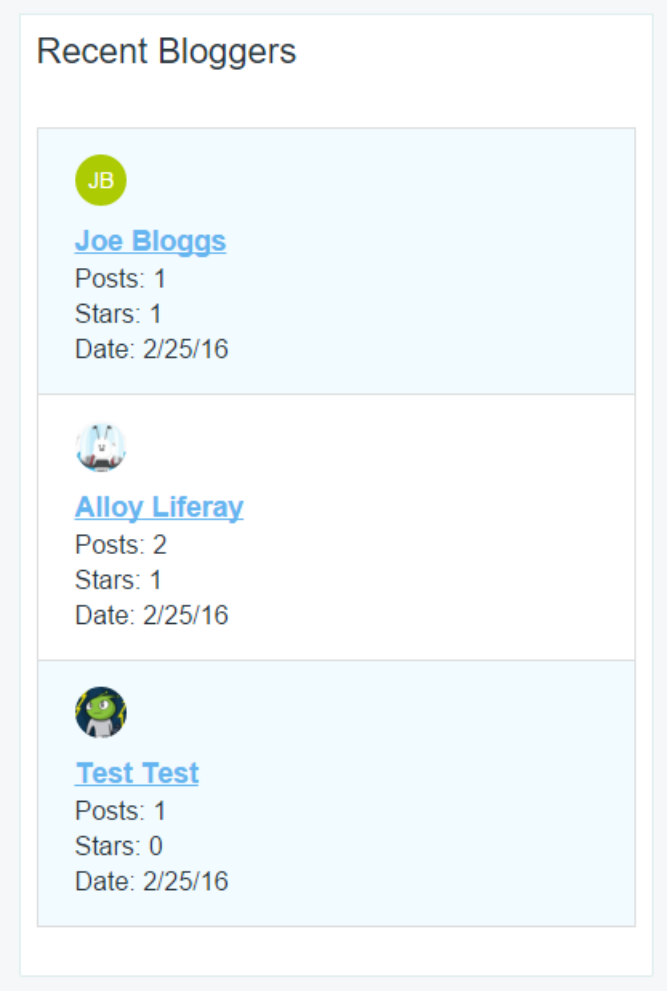


Figure 19.8: You can show off your site or organization's most recent bloggers from the Recent Bloggers app.

Message Boards offers many configuration options. They are straightforward to use and are the reason why it's a full-featured forum application for your site. Let's create a message board.

Creating Message Boards

You can create a message board instance for a Liferay DXP instance, a site, or page. An instance's message board can be used on any of its pages. A message board can be scoped to a site, including a regular site, an organization's site, or a user's profile site or dashboard site. Message Boards in Site Administration lets you administer all of a message board's options. Each Liferay DXP instance and site has a message board instance built-in. If you want to work with a message board scoped to a page, you must create the page scope from a Message Boards application on that page.

To scope a message board to a page, first add the Message Boards application to the page. Then click the app's *Options* icon ( ), select *Configuration*, and select the *Scope* tab, to list the available scopes. Select the scope with page's name, or *PageName (Create New)*, if the scope hasn't been created already. Your message board is now scoped to the page and available to work with in Site Administration.

To administer a message board, open the *Menu* ( ) and navigate to the *Content* section for your site, page, or global scope. The Message Boards administration screen appears.

Note: To administer a message board for a particular page scope, click the *Site Selector* button ( ) to list the available scopes, and select the scope with the page's name. Finally, select *Message Boards* from under the *Content* heading.

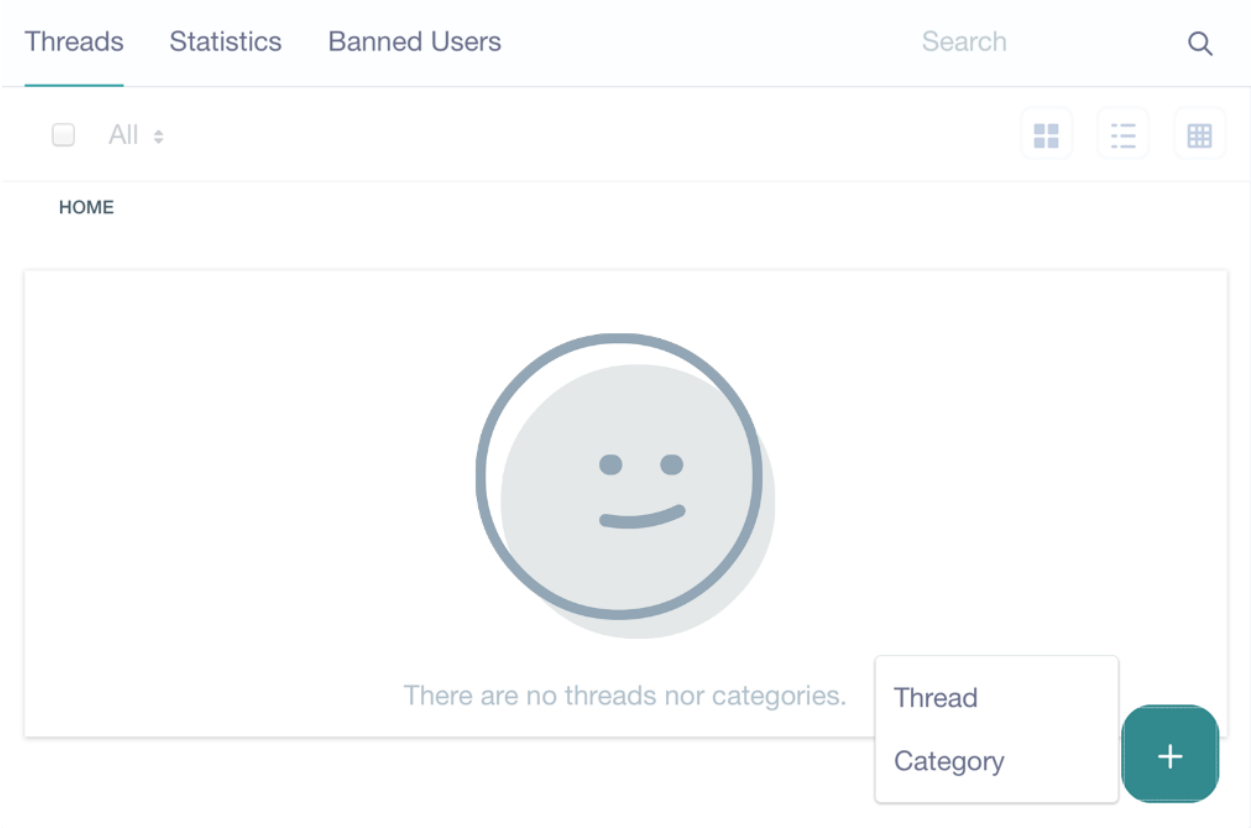


Figure 19.9: A Message Board instance starts empty, ready for you to configure for your purposes.

Initially, a message board has no categories or threads; and that's good, because you need to configure it for your purposes. To open the message board's configuration screen, click the message board's *Options* icon ( ) and select *Configuration*. From the Configuration tabs, you can configure the message board for the behavior you want.

General Setup From the *General* configuration tab, you can enable anonymous posting, subscribe by default, flags, ratings, and thread as question by default. You can also choose whether to use BBCode or HTML as your message format. You can also choose how long posts display on the message board. Anonymous posting, subscribe by default, flags, and ratings are selected by default and the default message format is BBCode.

Enabling *Allow Anonymous Posting* allows guest users to post messages to your message board. Whether or not you should do this depends on the type of community you are building. Allowing anonymous posting opens your site to anyone who might want to spam your forums with unwanted or off topic advertising messages. For this reason, most message boards administrators turn anonymous posting off by unchecking this box.

Enabling the *Subscribe by Default* option automatically subscribes users to threads they participate in. Whenever a message in a thread is added or updated, Liferay DXP sends a notification email to all users subscribed to the thread.

You can set the *Message Format* to either BBCode or HTML. This determines the markup language of users' actual message board posts. The type of WYSIWYG editor presented to users depends on which option is enabled. Both editors have a *Source* button which lets users view a message's underlying BBCode or HTML. Users can compose messages using either the WYSIWYG or Source view and can switch between views during message composition by clicking on the *Source* button.

Enabling *Enable Flags* lets users flag content which they consider to be objectionable. If you are allowing anonymous posting, you might use flags in combination with it if you have someone administering your message boards on a daily basis. That way, any unwanted messages can be flagged by your community, and you can review those flagged messages and take whatever action is necessary. Using flags is also a good practice even if you're not allowing anonymous posting.

Enabling *Enable Ratings* allows users to score posts. The scores are used by Liferay DXP's social activity system to rank site members by how helpful their contributions are. You can read more about social activity in the Measuring Social Activity article.

Enabling the *Thread as Question by Default* option automatically checks the mark as question box in the new thread window. Threads marked as questions display the flag "waiting for an answer." Subsequent replies to the original message can be marked as an answer.

Lastly, you can set the amount of time a post is displayed on the message board's *Recent Posts* tab until it is taken away. You have options of 24 hours, 7 days, 30 days, and 365 days. After the time has passed, the post is removed from *Recent Posts*. However, the post itself isn't deleted. It remains accessible everywhere else in the message board.

Email Setup Message Boards provides options for configuring the email sender and specifying the mail message to use when someone posts a new message board thread or updates a thread.

Email From: This tab allows you to configure the name and email address from which message board email notifications are sent. Default values, are from the `admin.email.from.name` and `admin.email.from.address` portal properties. These were set in the Basic Configuration Wizard when installing Liferay DXP. Make sure to update this email address to a valid one that can be dedicated to notifications. You can determine whether to support HTML in your email messages. Lastly, you can expand the *Definition of Terms* section to see variables available to use in your emails.

Message Added Email: This tab allows you to customize the email message that users receive when a message is added to a topic to which they are subscribed.

- **Enabled:** allows you to turn on the automatic emails to subscribed users. Uncheck the box to disable the message added emails.
- **Subject:** lets you choose a prefix to be prepended to the subject line of the email. This is usually done so that users can set up message filters to filter the notifications to a specific folder in their email clients.
- **Body:** allows you to write some content that should appear in the body of the email.
- **Signature:** lets you add some content to appear as the signature of the email.
- **Definition of Terms:** This section defines certain variables which you can use in the fields above to customize the email message. Some of these variables are for the message board category name, the site name, and more.

Message Updated Email: The Message Updated Email tab is identical to the Message Added Email tab, except it defines the email message that users receive whenever a message is updated. Next, you can specify priority classifications for message threads.

Thread Priorities You can define custom priorities for message threads on this tab. This allows privileged roles to tag a thread with a certain priority in order to highlight it for users. By default, three priorities are defined: Urgent, Sticky, and Announcement. To define a thread priority, enter its name, a URL to the image icon that represents it, and a priority number. Threads with a higher priority are posted above threads with a lower priority.

Thread Icons

Icon	Definition
	Urgent
	Announcement
	Sticky
	Question

The localized language field lets you name the priorities in each locale. You can select the locale, update the priority names for it, and save your updates.

User Ranks On this tab, users can be ranked according to the number of messages they have posted. You can set up custom ranks here. Defaults have been provided for you, going from zero messages all the way up to one thousand.

In addition to ranks, you can also choose labels for certain users to have displayed in their profiles as shown by the Message Boards application. The labels correspond to memberships the users have in Liferay DXP. Below are examples of using the label *Moderator*. The Moderator label in this configuration is applied for anyone who is a part of any of the Message Boards Administrator groups: the site role, the organization, the organization role, the regular role, or the user group. Of course, you probably wouldn't want to create a role, organization, organization role, site role, and user group all with the same name in your Virtual Instance, but you get the idea.

```
Moderator=organization:Message Boards Administrator

Moderator=organization-role:Message Boards Administrator

Moderator=regular-role:Message Boards Administrator

Moderator=site-role:Message Boards Administrator

Moderator=user-group:Message Boards Administrator
```

As you can see, all you need to do is set the rank, the collection type, and the name of the type. In the example above, anyone who has a site role, an organization role, a regular role, or is in a user group called *Message Boards Administrator*, or anyone who is the organization owner gets the Moderator rank.

As with thread priority names, the *Localized Language* field lets you localize rank names. RSS feed configuration is next.

RSS Message board threads can be published as RSS feeds. This tab allows you to enable/disable RSS subscriptions and define how the feeds are generated.

Maximum Items to Display: lets you select the number of items to display in the feed.

Display Style: lets you select the style. You can publish the full content, an abstract, or just the title of a thread.

Format: allows you to choose the format: RSS 1.0, RSS 2.0, or Atom 1.0.

Once you've finished configuring your message board, make sure to *Save* your changes.

From Message Boards in Site Administration, you can configure the message board's permissions.

Message Board Permissions To edit message board permissions, click on the *Options* icon ( ) and select the *Home Category Permissions* option. This permissions screen allows you to grant and revoke a role's ability to access parts of the application and perform particular actions.

The permissions enable a role to perform the following actions.

Lock Thread: Stop any further additions or modifications to a thread's messages.

Subscribe: Enlist yourself to receive notifications on new and modified posts.

Reply to Message: Respond to an existing message.

Add File: The user can attach a file to his message.

Permissions: View and modify permissions.

Add Message: Post a new thread.

View: View all the contents of message threads.

Add Category: Add a new category to the message board.

Update Thread Priority: Modify a thread's priority.

Ban User: Forbid a user from participating in the message board.

Move Thread: Move a thread to a different category or subcategory.

Configure the roles with the permissions you want and *Save* your changes.

In addition to the *Home Category Permissions* option, there the *Permissions* option. It's for specifying the application instance's general permissions. It lets you control access to the application instance's options menu and to that menu's Configuration and Permissions options. There's also Preferences permission that allows you to control access to the application instance's preference options, if the app has any custom preferences.

Now that you've configured your message board and set its permissions, you can focus on adding categories for message board threads.

Adding Categories

You are now ready to add categories to your message boards. Click the *Add* icon ( ) and select *Category* button. Enter a name for the category and a description of the category.

Categories can have different display styles. The available categories must be set in portal property `message.boards.category.display.styles` and the default category in `message.boards.category.display.styles.default`. When creating a new category, you can select the display style you like. By default, Message Boards provides two predefined display styles, although many more can be easily added:

Default: classic display style for general purpose and discussions.

Question: designed for discussions in a format of questions and answers.

The Permissions section provides a selector for quickly choosing who can view the category and a link to a table that presents the category's other permission options. The *Viewable by* selector lets you choose from the following values to determine who can view the category: Anyone (Guest Role), Site Members, or Owner.

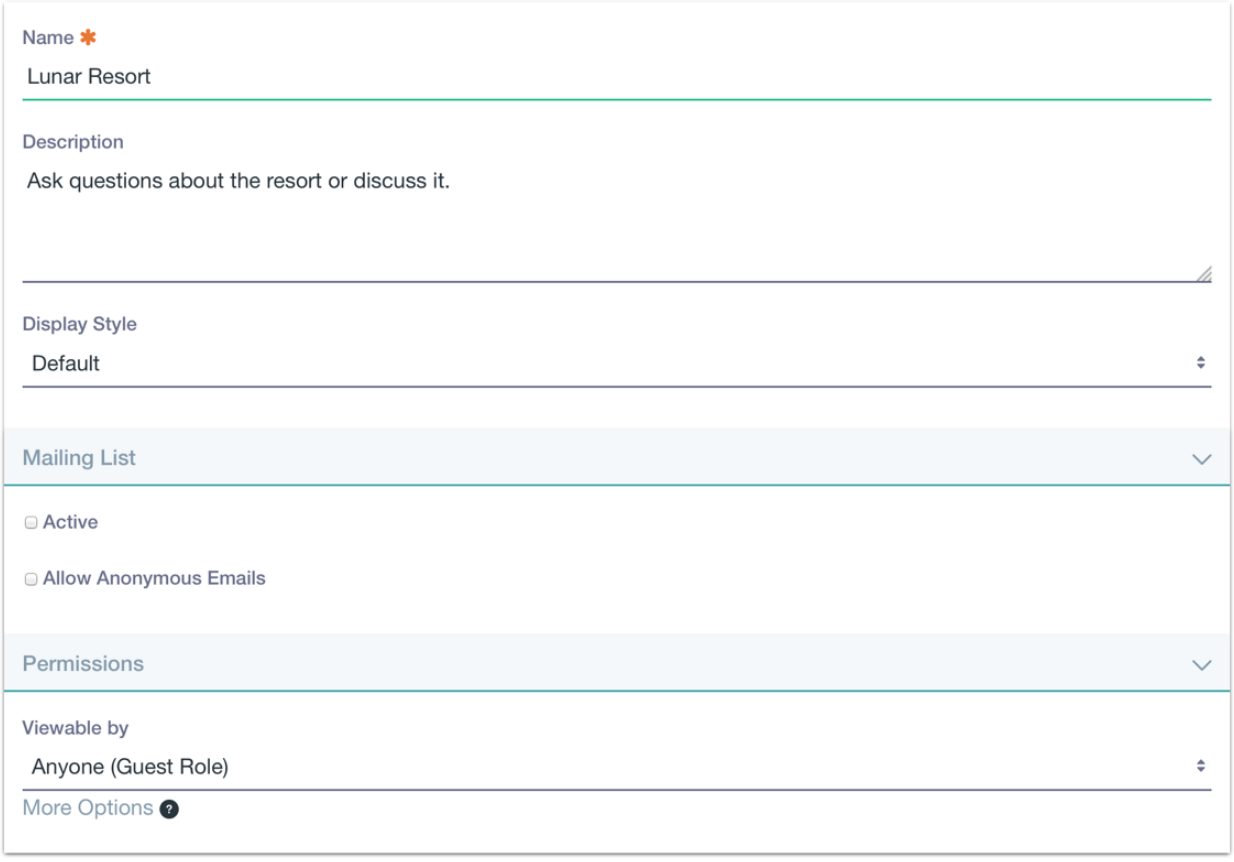


Figure 19.10: You have several options to create a message board category for your needs.

To show more permission options, click *More Options*. A table appears with the rest of the category's permissions:

Lock Thread: Stop any further additions or modifications to a thread's messages.
Add Subcategory: Add a new category within this category.
Update: Edit the category.
Subscribe: Enlist yourself to receive notifications on new and modified posts.
Reply to Message: Respond to existing messages.
Add File: Attach a file to any of your messages.
Permissions: View and modify permissions.
Delete: Remove the category.
Add Message: Post a new thread.
Update Thread Priority: Modify a thread's priority.
Move Thread: Move a thread to a different category or subcategory.

After you've created a category, you can revisit its permission options by clicking the category's *Actions* icon (⋮) and selecting *Permissions*.

The form also lets you enable the mailing list function. If don't want to add a mailing list to the category you're creating, you can save your changes now. You can always edit an existing category to add, edit, or remove a mailing list. The next section explains user subscriptions and mailing lists.

You can add as many categories to your message boards as you wish. And categories can have subcategories. You can add any number of top-level categories to a message board. You can also edit a category and add subcategories to an unlimited level. For usability reasons, you don't want to nest categories too deep, or your users will have trouble finding them. You can always add more categories as your message boards grow. Finally, each category can have any number of threads.

Also, you may move a category or merge a category's threads with a different Parent Category. To bring up the the Move screen, select the category's *Move* option. You can select a different Parent Category, optionally select the *Merge with Parent Category* check box, and click *Move*.

As you add categories to a message board, they're listed on the message board's home. The list displays the names of the categories and the numbers of subcategories, threads, and posts in each one. To add a subcategory to category, click on the category's name in the list and then click the *Add* icon and select *Subcategory*. The form for adding a subcategory appears and is populated, by default, with the properties of the parent category. This includes the parent category's display style (Default or Question) and mailing list configuration. Of course, you can change the display style or mailing list configuration of a subcategory just as with any category. The *Add Subcategory* form and the *Add Category* form are the same.

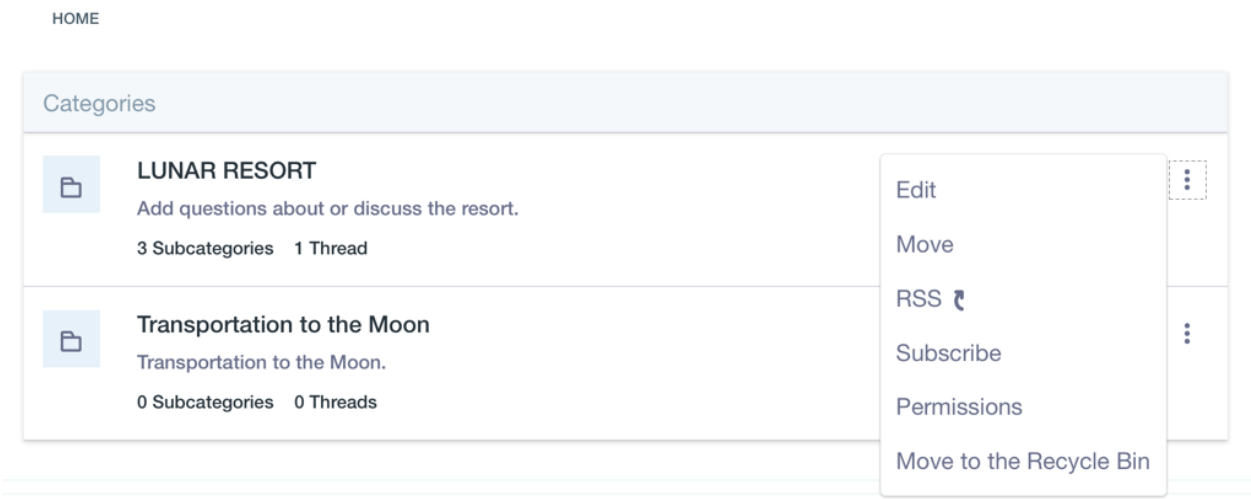


Figure 19.11: Categories help you organize threads so users can find topical threads that interest them.

Liferay DXP's Message Boards supports two different mechanisms for sending email notifications: user subscriptions and mailing lists. Let's discuss user subscriptions first and then mailing lists.

User Subscriptions and Mailing Lists

The first mechanism Liferay DXP uses for sending email notifications is user subscriptions. Users can subscribe to particular categories and threads. Liferay uses the message board's configured *Email From* address to send email notifications to subscribed users, whenever a new post is created or an existing post is updated. Liferay can import email replies to message board notifications directly into the message board. This is a very useful feature since it allows users to interact on the message board via email, without needing to log in to Liferay DXP and view the message board page directly. This feature is, however, disabled by default. To enable it, add the following line to your `portal-ext.properties` file:

```
pop.server.notifications.enabled=true
```

As this property suggests, Message Board's user subscription mechanism uses the POP mail protocol. When an email reply to a message board notification is read by Liferay DXP, the reply is posted to the message board and then deleted from the mail server. Deleting the message from the mail server is the POP protocol's default behavior and Liferay DXP assumes that your POP mail server behaves this way. Most POP clients offer an option to leave mail on the mail server after it's been downloaded, but you shouldn't exercise this option. If you configure mail to be left on the mail server, Liferay DXP will repeatedly send copies of each retained message along with each new email notification that's sent to subscribed users.

When enabling message boards to import replies to email notifications, you should decide whether or not you want to use a mail server subdomain to handle notifications. By default the following property setting is specified in the portal properties:

```
pop.server.subdomain=events
```

This property creates a special MX (mail exchange) subdomain to receive all virtual instance related email (e.g., events.liferay.com). If you don't want to use the subdomain approach, you can unset this value to tell Liferay to use the *Email From* address, specified in the Message Board's configuration, to receive message board notification email replies. For example, the *Email From* address could be set to *replies@example.com*.

If you don't want to use a mail server subdomain, add the following line to your `portal-ext.properties` file:

```
pop.server.subdomain=
```

If you're not using a mail subdomain, Liferay DXP parses the message headers of emails from the *Email From* address to determine the message board category and message ID. If you keep the `pop.server.subdomain=events` default, the email notification address takes the following form: *mb.[category_id][message_id]Liferay DXPproduct@ parses the email address to find the category and message ID. Parsing the email address is safer than parsing message headers, since different email clients treat message headers differently. This is why the events subdomain is enabled by default.

Additionally, you can configure the interval on which the `POPNotificationListener` runs. The value is set in one minute increments. The default setting is to check for new mail every minute, but you can set it to whatever you like:

```
pop.server.notifications.interval=1
```

Note: Depending on your mail provider, if you're using multiple devices to access email through POP, you might need to configure in your POP settings something like Gmail's *recent mode*. In Gmail, recent mode assures that emails go to all your devices instead of only the the first client that receives the email. To enable recent mode in Gmail, for example, prefix the value of your POP client's Username or Email field with `recent:`.

The second mechanism Liferay DXP uses for sending email notifications is mailing lists. Any category in a Liferay DXP message board can have its own mailing list. Liferay DXP's mailing list mechanism, unlike its user subscription mechanism, supports both the POP and the IMAP protocols. POP is the default protocol, but each message board's mailing list is configured independently. If you choose the IMAP protocol for a category's mailing list, make sure to configure the IMAP inbox to delete messages as they are pulled by the email client that sends messages to the users on the mailing list. Otherwise, each email message that's retained on the server will be sent to the mailing list each time there's a new post or an update in the category.

When a mailing list is enabled for a message board category, Liferay DXP listens to the specific email inbox that's configured for the mailing list. Enabling the mailing list function allows users on the mailing

list to simply reply to the notification messages in their email clients. Liferay DXP pulls the messages from the email inbox it's configured to listen to and automatically copies those replies to the appropriate message board thread.

With both user subscriptions and mailing lists, users can reply to message board notification emails and Liferay DXP imports their replies to the message board. However, with mailing lists, users reply to the mailing list and Liferay DXP listens to the specific inbox configured for the mailing list and copies messages to the appropriate message board category. With user subscriptions, by default, email replies to message board notifications are not imported to the message boards. This feature has to be enabled in your `portal-ext.properties` file. Once this feature has been enabled, users can reply to a specific address and have their replies copied to the message board.

Note: Since any number of sites can use a globally scoped message board, globally scoped message boards do not support user subscriptions or mailing lists. Make sure to use a site-scoped or page-scoped message board if you need user subscriptions or a mailing list with your message board.

To enable the mailing list functionality for a category, you need a dedicated email address for the category. Then, click on your category's *Edit* option. Once you select the *Active* check box, a number of other options appear. When a mailing list is activated, Liferay DXP imports messages it receives from the mailing list to the message board. Liferay DXP looks for a Liferay user with the sender's email address. If the sender isn't a Liferay user and the *Allow Anonymous Emails* box is unchecked, the message is thrown away and not posted to the message board. If the *Allow Anonymous Emails* box is checked, anyone can send email to the message board category's dedicated email account and Liferay DXP copies the messages to the message board.

Email Address: lets you enter the email address of the account that will receive the messages.

Next, there are two sections: *Incoming* and *Outgoing*. These define the mail settings for receiving mail and for sending mail. The Incoming section has the following options:

Protocol: lets you select POP or IMAP.

Server Name: lets you enter the host name of the mail server you are using.

Server Port: allows you to specify the port on which your mail service is running.

Use a Secure Network Connection: lets you use an encrypted connection if your server supports it.

User Name: lets you enter the login name on the mail server.

Password: lets you enter the password for the account on the server.

Read Interval (Minutes): allows you to specify how often Liferay DXP polls the server looking for new messages to post to the message board.

The Outgoing section has the following options:

Email Address: lets you enter the email address that messages from this category should come from. If you want your users to be able to reply to the categories using email, this should be the same address configured on the *Incoming* section.

Use Custom Outgoing Server: allows you to use a different mail server than the one that is configured for the Liferay DXP instance. If you check this box, more options appear:

- **Server Name:** lets you enter the host name of the SMTP mail server you are using.
- **Server Port:** allows you to specify the port on which your mail service is running.
- **Use a Secure Network Connection:** allows you to use an encrypted connection if your server supports it.
- **User Name:** lets you enter the login name on the mail server.
- **Password:** lets you enter the password for the account on the mail server.

When you're finished configuring the mailing list for your category, click *Save*.

Now that you've created message boards and message board categories, you can explore posting messages to them and interacting with other user's threads.

Using the Message Boards

You can add the Message Boards application to a page from the *Add* ( ) menu's *Collaboration* section. Users will immediately recognize that the interface is similar to many other implementations they've seen before. Message boards are nothing new to the Internet, and many people have been using them for quite a long time. In any case, it can't hurt to explore how to use Liferay Message Boards and discover all of its features.

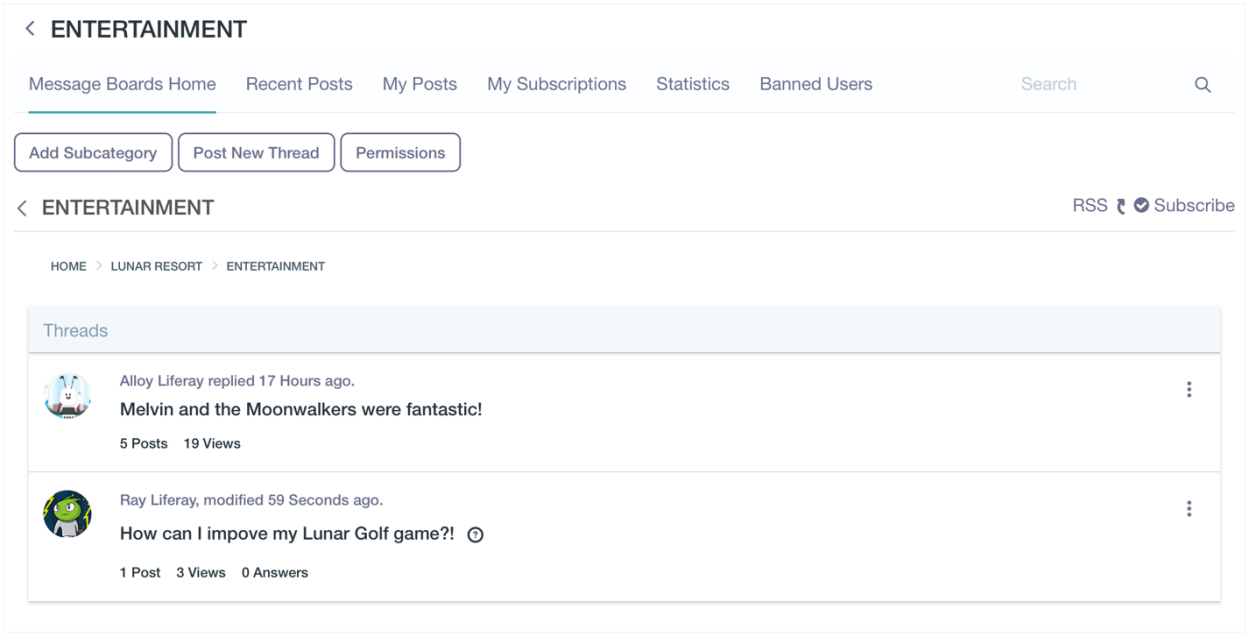


Figure 19.12: The Message Boards application lets you explore its categories, interact with message threads, and post new messages.

Threads can be viewed in many ways. At the top of the application is a set of tabs: *Message Boards Home*, *Recent posts*, *My Posts*, *My Subscriptions*, and for administrative users, *Statistics* and *Banned Users*. The Recent Posts tab shows all posts from all categories by date, so you can keep up on all the most recent discussions in the message boards. The My Posts tab shows all of the user's posts. It provides a convenient interface for revisiting previous conversations in order to retrieve pertinent information. The My Subscriptions tab allows the user to manage thread subscriptions. If you lose interest in a particular topic, you may want to visit this tab and unsubscribe from a category or thread.

For administrators, the Statistics tab shows the number of categories, the number of posts, and the number of participants in your message boards. It also has a list of your message board's top contributors. The Banned Users tab shows all of the users who have been banned from posting on the message boards.

Posting New Threads To post a new thread click the *Post New Thread* button in the app or in Message Boards in Site Administration) and select *Thread*. A message editing form appears. The body field on this form is different from that of the other Liferay DXP applications. The reason for this is to support *BBCode*, which is a standard form of markup used in many message board products. Before BBCode was invented,

many message board products would allow users to enter HTML to format their messages. This, however, enabled attackers to insert malicious code into the message board. BBCode was invented to provide users a way of formatting their messages without allowing them to enter HTML. Similarly, Liferay supports BBCode in the message boards because the other editors–which are used for the Content Management System, Blogs, and other applications–produce HTML. This is appropriate for those other apps, as they are only used by privileged users, but it is not appropriate for the message boards. Besides this, many users of message boards are familiar with BBCode and are used to it, and the editor that is provided for Liferay's Message Boards application makes it very easy to use.

Message Boards uses a rich-text editor. It supports bold, italicized, underlined, and crossed-out text, links, images, colors, lists, tables, alignments, quotation blocks, code blocks, different fonts and font sizes, and more. The editor has keyboard shortcuts Ctrl+b for bold, Ctrl+i for italics, and Ctrl+u for underline. You can mention another other user by entering the "@" character followed by the user's user name. There are even a bunch of smiley faces that you can use.

After entering the message's *Subject*, enter your message *Body*. Below the Body are sections that let you attach files (e.g., images) to your message, categorize it, tag it, and relate it to other assets. expand the *More Settings* section in the list of sections below the content editor. The user can specify thread options that the message board permits. The options include marking the message as a question, posting anonymously, subscribing to the message thread, assigning the message a pre-defined priority, and allowing pingbacks to the message.

The user can also specify permissions. The message can be set to be viewable by a particular role. And the following additional permissions can be set on the message:

Update: Edit the message.

Subscribe: Receive notifications of updates to the message and its thread.

Permissions: Grant/revoke permissions for the message.

Delete: Remove the message.

View: View the message.

A permissions icon appears above each posted message. You can revisit the above permissions by clicking the permissions icon.

When you're done editing and configuring your message, you can preview it, and save it as a draft or publish it. Once it's published, it's listed along with the other threads in the category.

Message Boards is also highly integrated with Liferay DXP's user management features. When you click on a thread or thread reply's subject link, the author's profile picture, name, rank, number of posts, the date the user joined the site, and a link to the user's recent posts.

Participating in Message Board Threads To find message board threads that interest you, browse a message board's categories or Recent Posts. You can view a category's thread listing by clicking on the category's name. From within a category screen, you can subscribe to an RSS feed and/or emails that inform you about thread activities in that category. Similar to a category's screen, Recent Posts lists threads too, except they're the latest threads across all the categories.

To view a message thread, click on it. Messages are shown in a threaded view so that replies are aligned under their proper parent thread. This makes it easy to follow along with conversations. Thread replies are indented under their parent thread.

Subscribing to a thread causes Liferay DXP to send the user an email whenever a new message is posted to the thread. If you have enabled the mailing list feature for the category in which the thread resides, users can simply reply to these messages in order to post back to the thread, without having to visit your site.

Most threads get more interesting as users reply to them. You can start creating a response by clicking one of the following buttons: *Reply*, *Reply with Quote*, or *Quick Reply*. The *Reply* screen includes the same

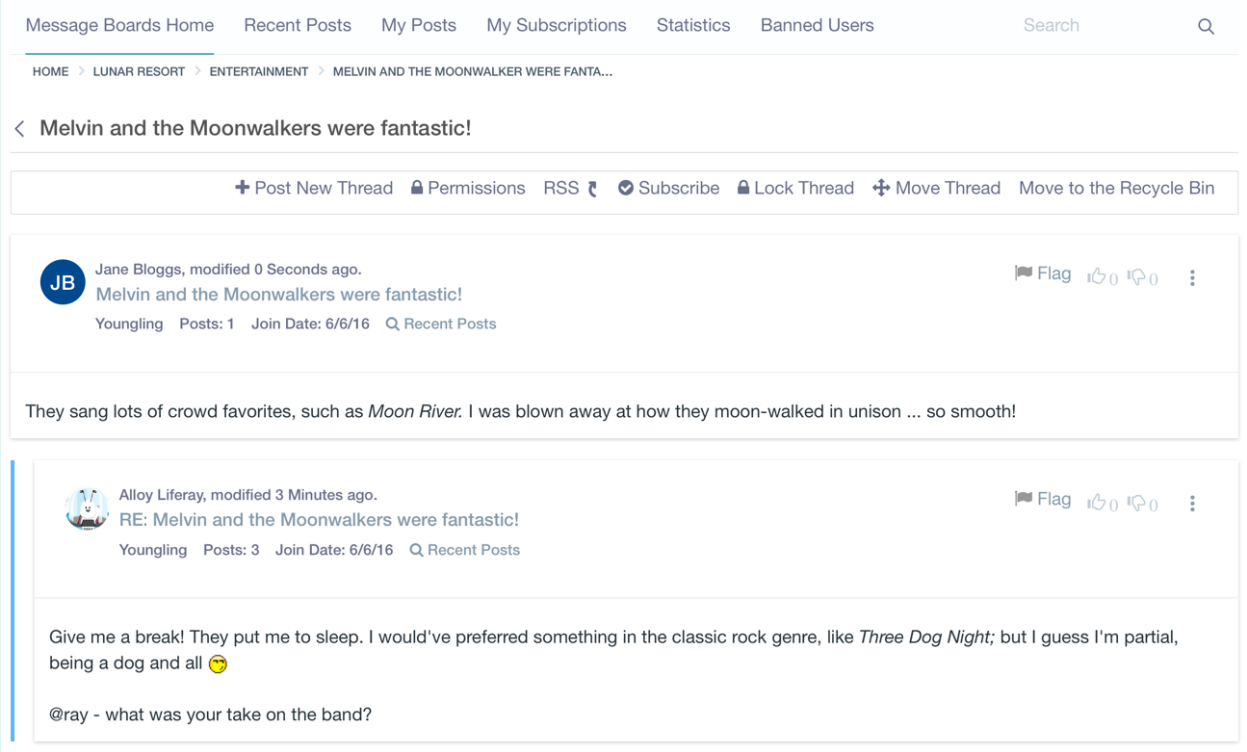


Figure 19.13: A thread's view displays author information and thread content, for the thread and all replies to the thread.

rich-text editor and options that you use for posting new messages. If you want to include text from the original message, click *Reply with Quote*. If you only need the rich-text editor to generate your reponse and don't need any other message options, click *Quick Reply*. The editor lets you insert emoticons, preformatted text, and more.

In addition to replying to a message, you can rate it or flag it as objectionable. A message board moderator can evaluate flagged messages and decide how to handle the messages and their authors. This provides an appropriate segue into the next topic: managing message boards.

Managing Message Boards

Message boards are powerful, but can become unweildy if left unmanaged. The Message Boards in Site Administration facilitates day to day thread administration. You may wish to separate this function out by a role, and then delegate that role to one or more users. That would free you up to concentrate on other areas of your site. To do this, you can, for example, create a role called *Message Board Administrator*. This role can be scoped to the Liferay instance, an organization, or a site. If you create an instance-scoped role, members of this role will be able to administer Message Boards throughout Liferay DXP. If it is an organization or site-scoped role, members of this role will be able to administer a Message Boards application in only that organization or site.

You can create such a role from the Control Panel. To define the role's permissions, click its *Define Permissions* action and navigate to *Site Administration → Content → Message Boards*. A screen appears that allows you to configure the various Message Boards permissions.

Select the permissions you would like message board administrators to have and save them. You can add users to this role. Message board administrators can perform all the message board functions already

presented, including creating and deleting categories and posting threads. In addition to these, a number of other functions are available.

Locking Threads You may encounter threads that you think should be preserved, but stoppped. You can halt activity on a thread by clicking *Lock Thread*.

Moving Threads Many times a user will post a thread in the wrong category. Administrators may in this case want to move a thread to the proper category. This is very easy to do. You can select the *Actions* menu (⋮) to the right of the thread and choose *Move*. Or, if you are already viewing the thread and you have administrative access, there is a link at the top of the thread labeled *Move Thread*. Click this link. You will be presented with a simple form which allows you to select a category to which to move the thread and a check box which allows you to post a message explaining why the thread was moved. This message will be posted as a reply to the thread you are moving. When finished, click the *Move Thread* button and the thread will be moved.

Deleting Threads Users with administrative access to the message boards can delete threads. Sometimes users begin discussing topics that are inappropriate or that reveal confidential information. In this case, you can simply delete the thread from the message boards. This is easy to do. First, view the list of threads. Click *Actions* button and select *Move to the Recycle Bin* to delete the thread. This does not prevent users from re-posting the information, so you may need to be vigilant in deleting threads or consider the next option.

Banning Users Unfortunately, sometimes certain users become abusive. If you wind up with a user like this, you can certainly make attempts to warn him or her that the behavior he or she is displaying is unacceptable. If this does not work, you can ban the user from posting on the message boards.

Again, this is very easy to do. Find any post which was written by the abusive user. Underneath the user's name/profile picture is a link called *Ban this User*. Click this link to ban the user from the message boards.

If after taking this action the user apologizes and agrees to stop his or her abusive behavior, you can choose to reinstate the user. To do this, click the *Banned Users* tab at the top of the message board. This will show a list of all banned users. Find the user in the list and select *Unban this User*.

Splitting Threads Sometimes a thread will go on for a while and the discussion completely changes into something else. In this case, you can split the thread where the discussion diverges and create a whole new thread for the new topic. Administrative users will see a *Split Thread* link on each post. To split the thread, click the link. You will be brought to a form which allows you to add an explanation post to the split thread. Click *OK* to split the thread.

Editing Posts Administrative users can edit anyone's posts, not just their own. Sometimes users will post links to copyrighted material or unsuitable pictures. You can edit these posts, which allows you to redact information that should not be posted or to censor profanity that is not allowed on your message boards. You can also update the thread's priority or mark a reply as an answer to a thread's question.

Permissions Permissions can be set not only on threads, but also on individual posts. You can choose to limit a particular conversation or a post to only a select group of people. To do this, click the *Permissions* link on the post and then select among the *Delete, Permissions, Subscribe, Update, and View* permissions for the particular role to which you want to grant particular access. This function can be used, for example, to allow some privileged users to post on a certain thread, while others are only allowed to view it. Other combinations of the above permissions are also possible.

That wraps up message board management basics.

Summary

As you've discovered, Liferay's Message Boards provides full-featured forums for users to ask questions, discuss topics, and share small amounts of information. Message boards help to build strong site communities. Next, let's learn how users can *mention* other users in their content and in comments and thread replies.

19.3 Mentioning Users

Have you ever wanted to include another user in a discussion on the Message Boards? Have you ever wanted to give kudos to a colleague in content you're writing? With the Mentions feature, you can *mention* (notify and/or draw attention to) other users, by entering the "@" character in front of each user's user name.

When you mention a user, the user receives a site notification next to the user's profile icon and an email, alerting the user with a link to the content. You can mention users in a blog entry, a message boards thread or thread reply, or comments in any app that supports comments. What's more, a mention in app content links to the user's home page, so readers can find out more about the mentioned user.

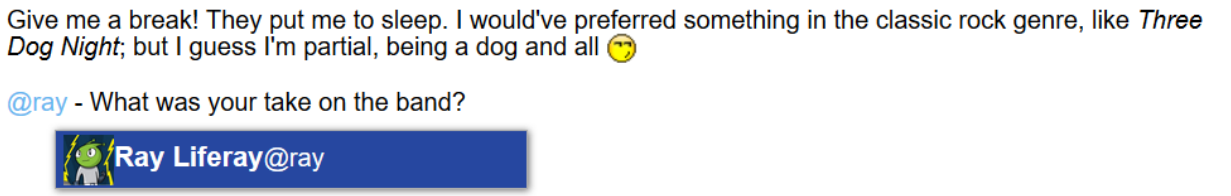


Figure 19.14: As you enter a user name after an "@" character, Mentions displays links to users that match the text you enter. Select the user you want to mention and publish your content.

The Mentions feature works as you'd expect. While editing comments or content, enter an "@" character followed by the *user name* of the user you're mentioning (e.g., *@joe.bloggs*). A selector appears, listing users that match the name you're entering. In the selector, each user is represented by his/her profile picture, name, and user name. Click on the user you want to mention and finish editing your content.

Known Limitation: In comments, homepage hyperlinks aren't displayed for mentioned users. Refer to issue LPS-56020 for details.

On publishing the content, mentioned users receive a notification next to their profile picture and an email, informing them that they've been mentioned. The notification and email indicate the author's name and content type, and it links to the content.

Figure 19.15: Liferay DXP's Notifications feature alerts users with the number of notifications (including mentions) waiting for them to read.

Here's an example notification email message:

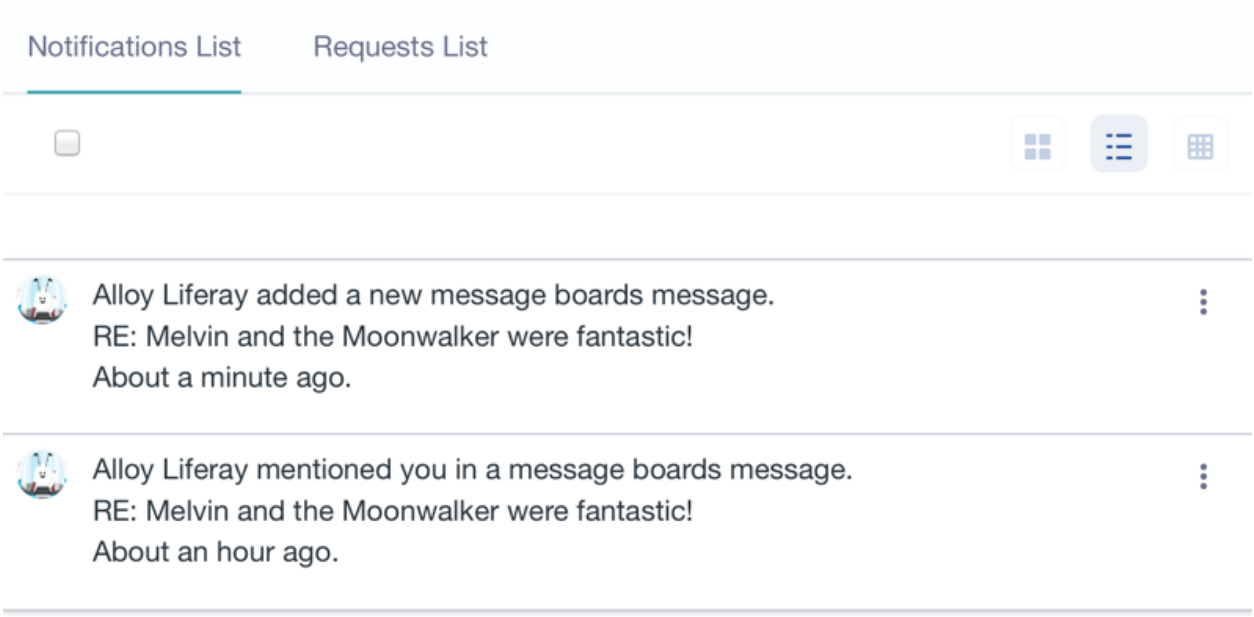


Figure 19.16: When you click on the notifications number next to your profile picture, your Notifications List appears.

```
Dear Joe Bloggs,

Neil Armstrong mentioned you in the following comment:

----------
I bet @joe would like visiting the moon.
----------

View comment.(link)

Sincerely,
Kim Admin
kim.admin@lunarresort.com
```

The Mentions app is a part of the Collaboration Suite. Mentions is enabled globally by default, but you can enable/disable it globally or per site. For a site to use Mentions, it must be enabled for the site's Virtual Instance. To access the global Mentions settings for your Virtual Instance, open the *Menu* (), then navigate to *Control Panel* → *Configuration* → *Instance Settings*, click on the *Social* tab, and expand the *Mentions* section.

All users are allowed to mention fellow site members and friends, by default. To fine tune these options, select the *Define Mentions Capability for Users* option and specify the settings you want.

Mentions must be enabled for a site's Virtual Instance in order for Mentions to be available for the site. As a site administrator, you can enable or disable Mentions for your site. A site's Mentions app configuration is accessible from within the *Menu* (). Once in the menu, navigate to *Site Name* → *Configuration* → *Site Settings*, click on the *Social* tab, and expand the *Mentions* section. This section lets you enable or disable Mentions for the site.

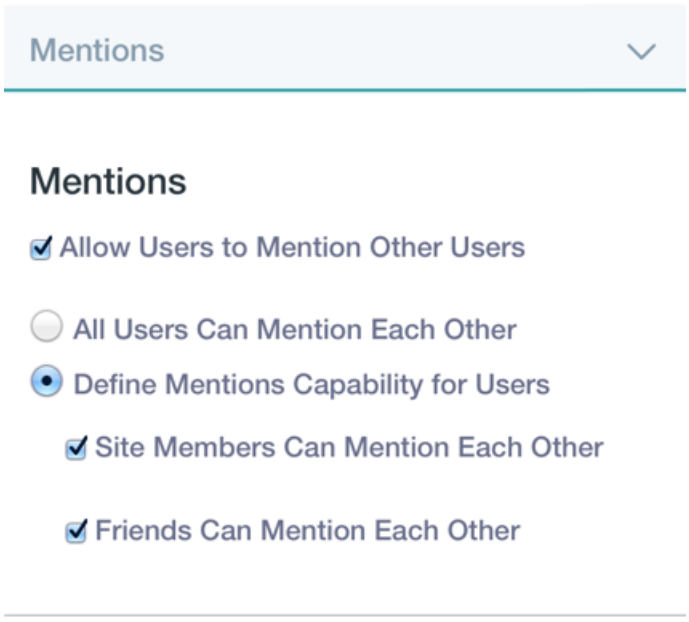


Figure 19.17: From Instance Settings in the Control Panel, you can enable or disable the Mentions feature for all of the Virtual Instance's sites.

As we've demonstrated, you can mention users to get their attention, praise them with kudos, and link to their home pages. It helps you welcome specific users to collaborate with you. Have fun collaborating and socializing with Mentions!

Now as your site's community gathers more information on topics or procedures relevant to users, you'll want to provide a means for them to collaborate on that information. Liferay DXP's Wiki is a terrific application for users to jointly produce hierarchies of information.

19.4 Working Together with the Wiki

Liferay DXP's Wiki is a full-featured wiki application which has all the features you would expect in a state of the art wiki. Again, though, it has the benefit of being able to take advantage of all the features of Liferay DXP. As such, it is completely integrated with Liferay DXP's user management, tagging, and security features.

So what is a wiki? A wiki is an application that allows users to collaboratively build a repository of information. There are, of course, many implementations of this idea, the most famous of which is Wikipedia. Wikipedia is a full online encyclopedia developed collaboratively by users from all over the world, using a wiki.

A wiki application allows users to create documents and link them to each other. To accomplish this, a special form of markup called wikitext is used. Unfortunately, the proliferation of many different wiki applications resulted in slightly different syntax for wikitext, as each new wiki tried to focus on new features that other wikis did not have. For that reason, a project called WikiCreole was started, which is an attempt to define a standard wiki markup that all wikis can support.

Rather than define another wikitext syntax, Liferay's Wiki app supports WikiCreole as its syntax. This syntax is a best-of-breed wiki syntax and should be familiar to users of other wikis. The app provides a handy cheat sheet for the syntax on the page editing form with a link to the full documentation if you wish to use some of WikiCreole's advanced features.

An instance of the Wiki application can be created for each scope: the entire Liferay DXP virtual instance (global scope), a site, or a page. It's time to see how to use your site's Wiki application instance.

Getting Started with Wikis

The menu provides the best place to start working with your wikis. To start working with wikis for your site, click on the *Menu* icon ( ), navigate to your site, and select the *Content* section. If you are updating an existing page-scoped wiki application instance you can select that page scope from the scope menu that the Gear icon ( ) makes available. The site's wiki application instance is available in the Default Scope. Once you're in the proper content scope click on *Wiki*. The Wiki administration screen appears. This screen allows you to add, modify, and delete wiki nodes. A Wiki application instance can contain many wiki nodes. By default, it contains one node, called *Main*.

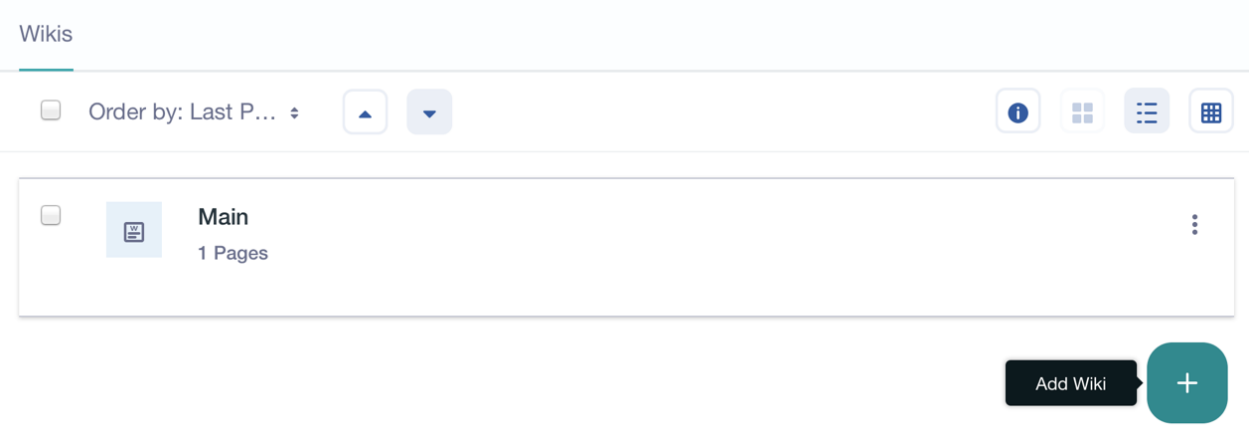


Figure 19.18: The Wiki application instance has a wiki node named *Main* with a single front page. You can build on the Main node or click the Add icon to create a new node.

Before you start adding to your wiki instance, you should configure it. The instance's interfaces for permissions, export and import, configuration, and application templates are accessible from the Options menu. Click the *Options* icon ( ) to open the menu.

Here are the wiki application instance options screens:

Wikis Permissions: allows you to specify which roles roles can create wiki nodes and which roles can access the Wikis Permissions screen.

Export / Import: enables you to import existing wiki content into your wiki application instance or export wiki content to a file. Refer to Importing/Exporting Pages and Content for details.

Configuration: has tabs for configuring email notifications and a tab for configuring RSS feeds. The *Email From*, *Page Added Email*, and *Page Updated Email* tabs are similar to other app's notification email settings tabs; they let you customize who wiki emails come from and the format and text of the email sent when a page is added or updated. The *RSS* tab allows you to configure RSS feeds.

Permissions: allows you to specify which roles can view the wiki application instance's options menu, access the menu's Configuration and Permissions options, and access any custom preference options added to the wiki application.

Select the *Wikis Permissions* option to open the Permissions screen. If you've created a specific role for creating wiki nodes and want to enable that role to create new wiki nodes in this wiki application instance, select the role's check box in the *Add Node* column and then click *Save*.

Now that we're done configuring our site's Wiki app instance, it's time to start working with a wiki node. Let's create a wiki node that everyone at the Lunar Resort can use to help build an encyclopedia on everything they know about space. We'll call it *Space Wiki*.

Click the *Add* icon ( ) to start creating the new wiki node. The New Wiki Node screen appears and prompts you to name and describe the wiki node. From this screen, you can also set up some default permissions. Click *Save* when you're done creating the wiki node.

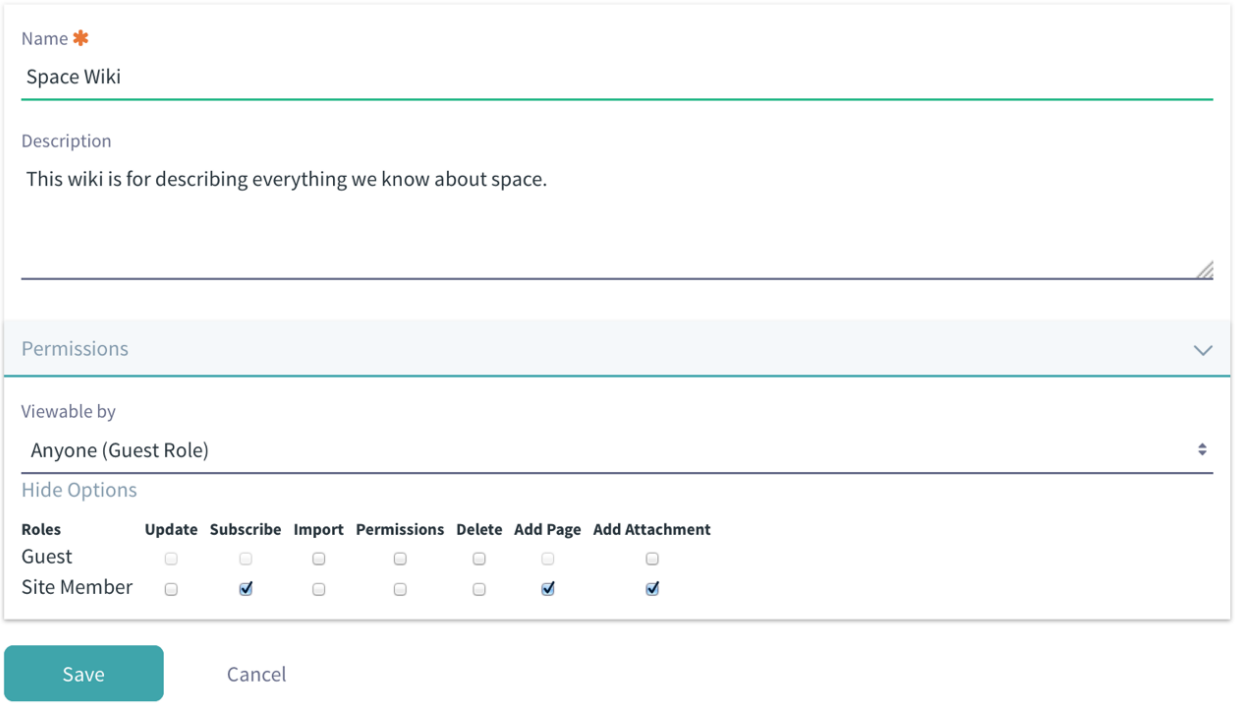


Figure 19.19: The New Wiki Node screen lets you describe your new node, set view permissions, and set permissions for Guest and Site Member roles.

On creating the new wiki node, you're brought back to the Wiki instance view. Your new wiki node accompanies the Main node in the list of this wiki instance's nodes. Next to each listed wiki node is an *Options* icon ( ).

Here are the wiki node options:

Edit: lets you edit the wiki's name and description.

Permissions: lets you specify which roles can add attachments to wiki pages, add pages, delete pages, import pages, set permissions on the wiki node, subscribe to modifications, update existing pages, and view the wiki node.

Import Pages: allows you to import data from other wikis. This lets you migrate off of another wiki application which you may be using and use the Liferay DXP wiki instead. You might wish to do this if you are migrating your site from a set of disparate applications (i.e., a separate forum, a separate wiki, a separate content management system) to Liferay DXP, which provides all of these features. Currently, MediaWiki is the only wiki that is supported, but others are likely to be supported in the future.

RSS: opens a new page where you can subscribe to an RSS feed using Live Bookmarks, Yahoo, Microsoft Outlook, or an application you can choose from your machine.

Subscribe: allows you to subscribe to a wiki node; any time a wiki page is added or updated, your Liferay DXP instance sends you an email informing you what happened.

View Removed Attachments: displays attachments that have been removed from the wiki node.
Move to the Recycle Bin: moves the wiki node to the Recycle Bin.
The figure below shows the wiki node options menu.

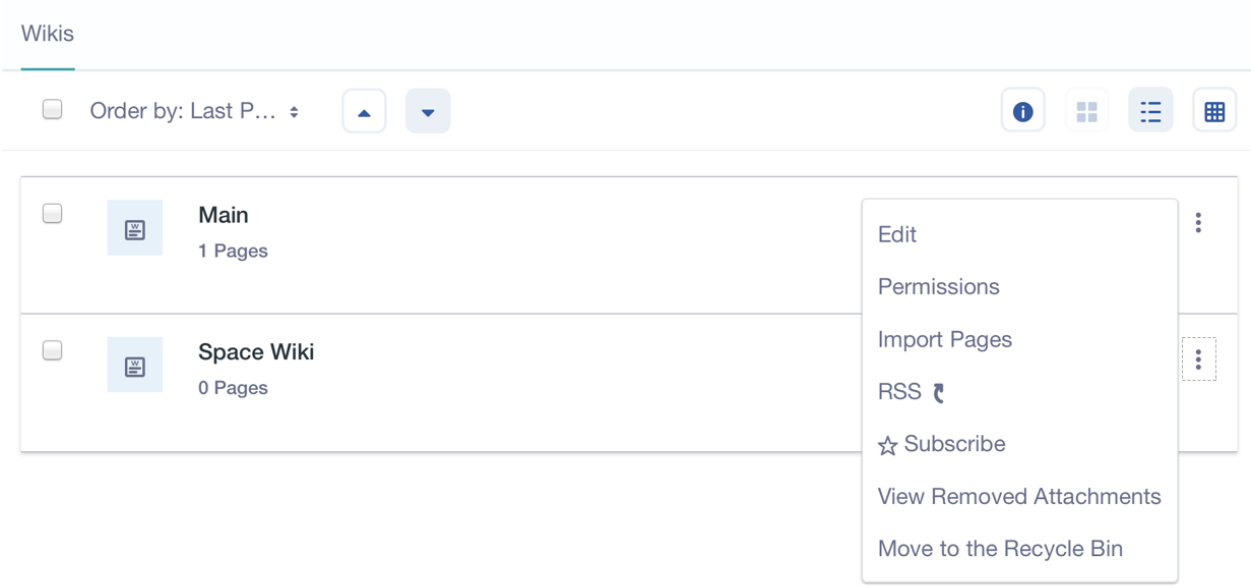


Figure 19.20: The Wiki application lists all of its wiki nodes. Each node's options icon menu lists node configuration screens and node actions you can perform.

Before you open up wiki nodes to contributors, you should consider whether to associate a workflow with them. For example, you could create a workflow that requires an administrator's approval to publish a wiki page modification (add, update, or delete). You can access your site's default *Wiki Page* workflow from within the Product Menu, by navigating to *Configuration → Workflow Configuration* for your site. To learn how to use workflow, refer to Using Workflow.

Next, let's create our wiki node's front page.

Adding and Editing Wiki Pages

When you create a wiki node, it has no pages. It's not until you navigate into that node that a default page called *FrontPage* is created automatically. To view the page, click on the wiki node's name. The FrontPage appears and shows a default message that explains the page is empty and needs you to add content. The message is a link; click on it to start editing the page.

Wiki uses a WYSIWYG editor, similar to the one used in Blogs. The content section's default message *Write your content here…* entices you to write. Click there and start entering text.

Next, highlight your pare of your text. Text formatting options appear. They let you markup the text as bold or italics, add it to an ordered or unordered list, or make it a link. You can also use keyboard shortcuts Ctrl+b for bold, Ctrl+i for italics, and Ctrl+u for underline. To remove all formatting, click the *Remove Format* icon. To insert an image, table, or horizontal line, you can click the + icon to bring up the insert content menu ().

If you need to modify an image for your wiki, use the Image Editor. Click the + icon and select the mountain silhouette to add an image. Select an existing image from the Documents and Media repository, and click the pencil icon () in the bottom right corner of the preview window, to open the Image Editor.

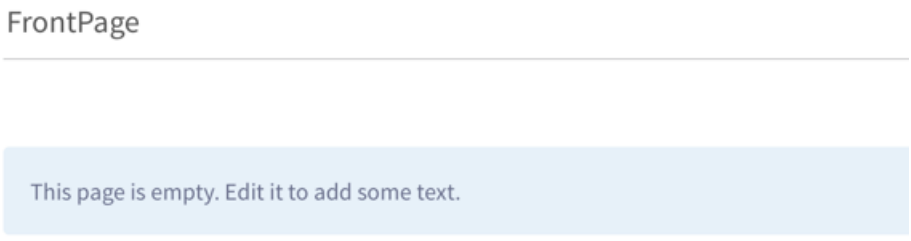


Figure 19.21: Each empty wiki page presents a default message link you can click to edit the page.

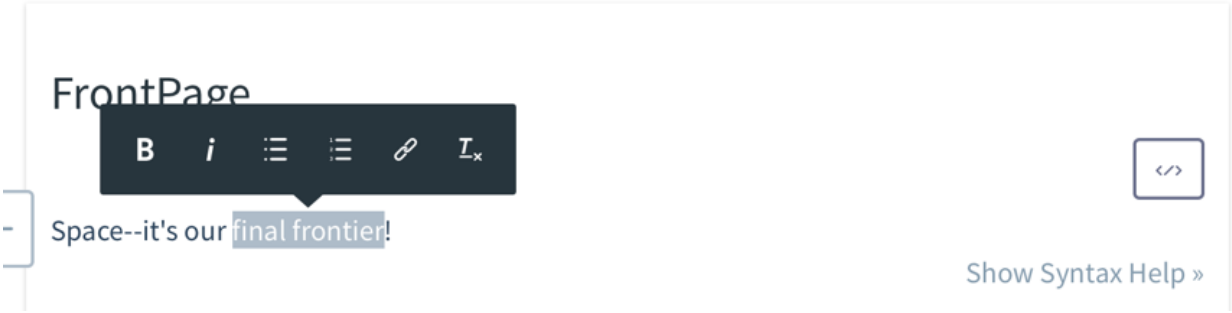


Figure 19.22: When you highlight text in the wiki page editor, a formatting toolbar appears. You can apply common markups to the text, make it a list item or a link, or remove existing markup.

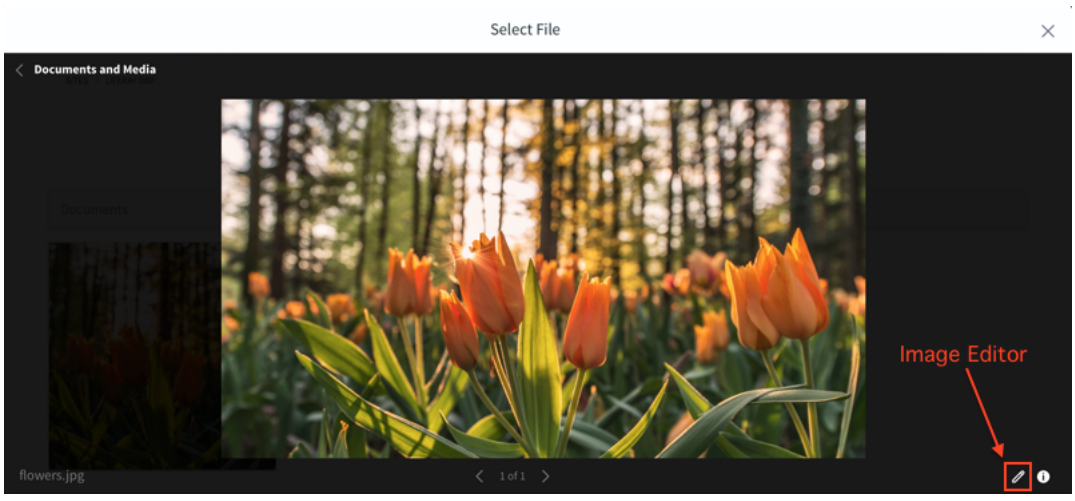


Figure 19.23: You can use the Image Editor to edit photos for your wiki.

Any edits you make are automatically applied to a copy of the image, which you can then use in your wiki.

You're working in the editor's regular mode. Source mode, on the other hand, lets you enter text in Creole format. You can switch to source mode by clicking the *Source* icon (</>) or switch back to regular mode by clicking the *Roller* icon ( )—the editor always displays the other mode's icon so you can switch between modes. Click on the *Source* icon to switch to source mode.

Notice that there's a convenient *Show Syntax Help »* link to a cheat sheet for helping you with the wiki syntax. You can use this syntax to format your wiki page content.

Consider, for example, the following wiki content written in Creole syntax:

```
Space--it's our //final frontier//! But we've learned so much about it, from our perspective at the Lunar Resort.

Let's share what we know about galaxies, solar systems, planets, moons, suns, and more!

[[Galaxies]]

[[Solar Systems]]

[[Planets]]

[[Moons]]

[[All kinds of stuff]]
```

Enter it as your page content. Then click on the *Enlarge* icon ( ) to edit the source in an enlarged dual screen editor. As you enter Creole format text in one pane the preview pane renders it. The editor lets you hide the preview pane, use a dark or light color source pane scheme, and arrange the panes horizontally or vertically.

At the bottom of the page editing screen, you can select *Categorization* to add tags. The tags link your wiki to categories. You can create categories from the Site Administration page, in the *Content → Categories* section. Categories are hierarchical lists of headings under which you can create wiki pages. This allows you to organize your content in a more formal fashion.

In the edit screen's *Configuration* section, you can set the page to use Creole wiki format, HTML, or plain text. We recommend that you stick with the Creole format, as it allows for a much cleaner separation of content and code.

Note: The MediaWiki and JSPWiki engines and formats are also available in respective deprecated apps on the Marketplace.

To install the JSPWiki engine, you must first remove the Creole engine by uninstalling the `com.liferay.wiki.engine.creole` bundle from your Liferay instance. The JSPWiki format will display in the format selector as *Creole* since it is an extension of the Creole engine.

Similar to other Liferay DXP applications, in the *Related Assets* section of the editor, you can select other assets to associate with the wiki page. And from the *Permissions* section, you can set general permissions for accessing and acting on the page.

Publish your page when you're done editing it. The next time you edit the page, the edit screen provides you with an *Attachments* section for attaching files to the page, via drag and drop or file upload.

When you're done editing in source view, click *Done*. When you're done editing the page, click *Publish*. The figure below shows the updated wiki page.

Did you notice that we added links for pages named *Galaxies*, *Solar Systems*, etc, that we haven't yet created? This is a common practice in building wikis–you add a link so that someone (maybe you) will create a page for it. The terrific thing is that you can create that new page for the link by clicking it. When you click a link to a non-existent page, the new wiki page screen appears for you to create the page. Liferay DXP

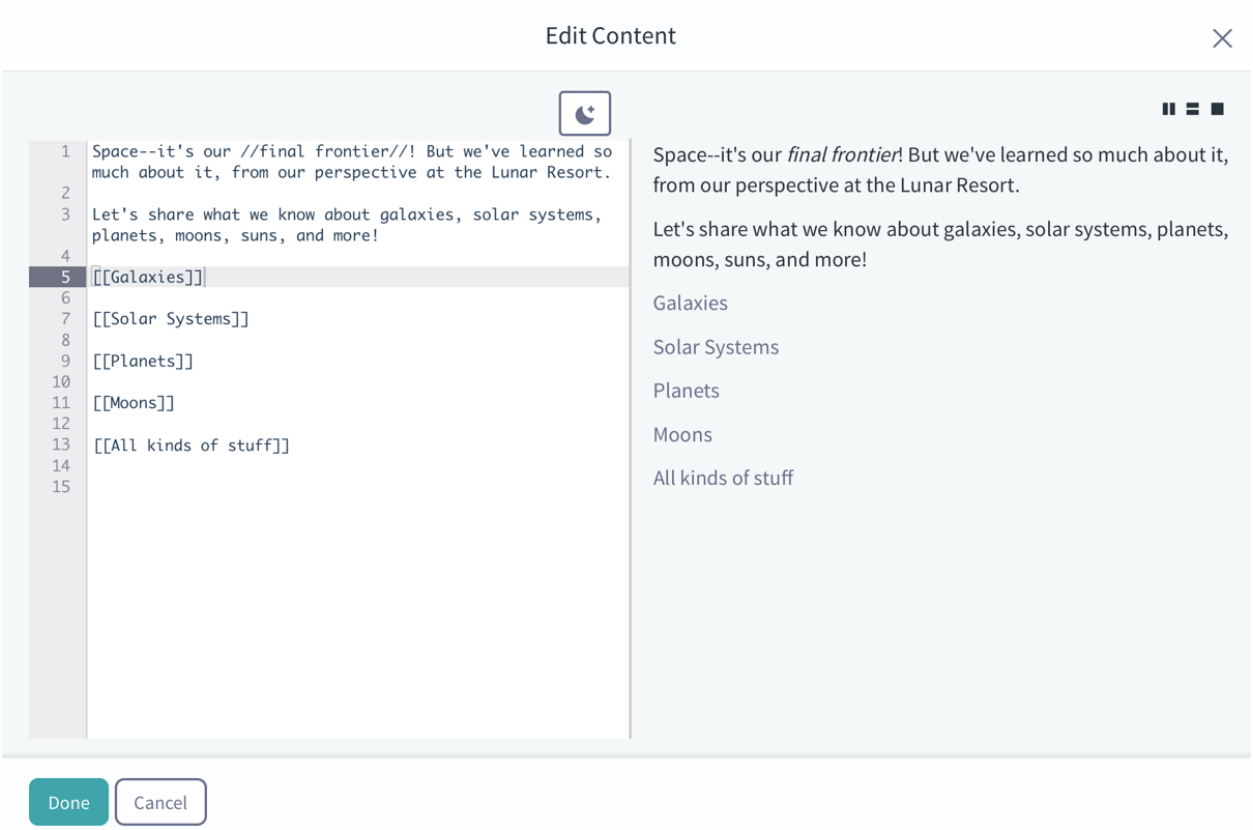


Figure 19.24: The wiki page source editor has a dual page mode for rendering content as you edit the source text.

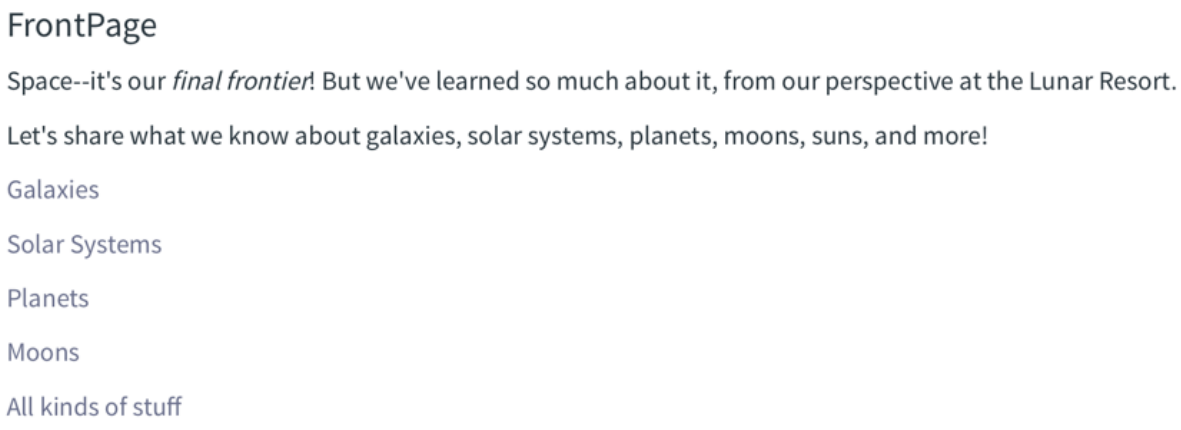


Figure 19.25: The Wiki's WYSIWYG editor helps you create attractive wiki pages.

displays a notice at the top of the page stating that the page does not exist yet, and that you are creating it right now. As you can see, it is very easy to create wiki pages. All you have to do is create a link, click on the link, and create the wiki page.

Note: when you create a page by clicking on a link to a page that does not yet exist, the new page is **not** a child of the current page. The page is created at the wiki node's root. From Wiki in Site Administration, you can use the page's Move action to assign it a new parent page. Clicking on the *Move* action brings up a window that lets you select a new parent for the wiki page.

On exiting the wiki page editor, you're brought back to the wiki node's view. It shows all of the node's top-level wiki pages. In our case, *FrontPage* is our only top-level page. If you navigate to a page that has child pages, its child pages are listed. In these page listings, each wiki page's Options icon ( ) lists the following options.

Edit: opens the wiki page in the wiki page editor.

Permissions: allows you to determine which roles can view, update, delete, subscribe to, or set permissions on the page, and add, update, or delete page discussions (comments).

Copy: brings up a wiki page editor window with all the content from the source wiki page. You're prompted to specify a new title for it.

Move: opens a dialog that allows you to rename the wiki page or assign the wiki page a new parent page within the wiki node.

Subscribe (or Unsubscribe): subscribes you to (or unsubscribes you from) notifications for the wiki page's modifications.

Move to the Recycle Bin: deletes the wiki page from the wiki node and moves it to the Recycle Bin.

Each wiki page has a check box next to it. When you select a page's check box, an Info icon ( ) and Recycle Bin icon ( ) appear in the top corner above the page list. To move the selected page to the Recycle Bin, click the Recycle Bin icon. To get additional information about the page, click the Info icon. The information menu provides a Star icon that you can select to subscribe to the page's modifications. The menu's Details section displays the page's Create Date and Last Modified date.

When you select multiple wiki pages, by selecting individual page check boxes or the Select All check box in the tool bar, the Info icon and Recycle Bin icon appear. They give you the wiki node's page count and let you move multiple pages to the Recycle Bin.

There are several more features in the wiki node view's toolbar. The *Order by:...* menu allows you to order the wiki pages by title or modification date. Next to that, the arrows let you arrange the pages in ascending or descending order. The display style options are next to the Info icon. To list the pages using a descriptive display style, click the *Descriptive* icon ( ). To list the pages using a list display style, click the *List* icon ( ).

As far as the other aspects of the Wiki application interface, we'll cover them as we use the Wiki application on site pages. Let's go to your site by clicking *Go to Site* and then close the product menu by clicking its icon ( ). We're ready to add Wiki applications to the site's pages.

Using the Wiki Applications on Site Pages

The Wiki application works just like the other Liferay applications. Open the Add menu by clicking the *Add* icon ( ). In the menu, navigate to *Applications* → *Wiki* and then click on *Wiki Add*.

Your site's wiki nodes display in the application. Click on the *Space Wiki* tab to view it in the app.

To view the Wiki application's configuration options, click on its *Options* icon ( ) and select *Configuration*. The Wiki app's Configuration screen appears and has several tabs.

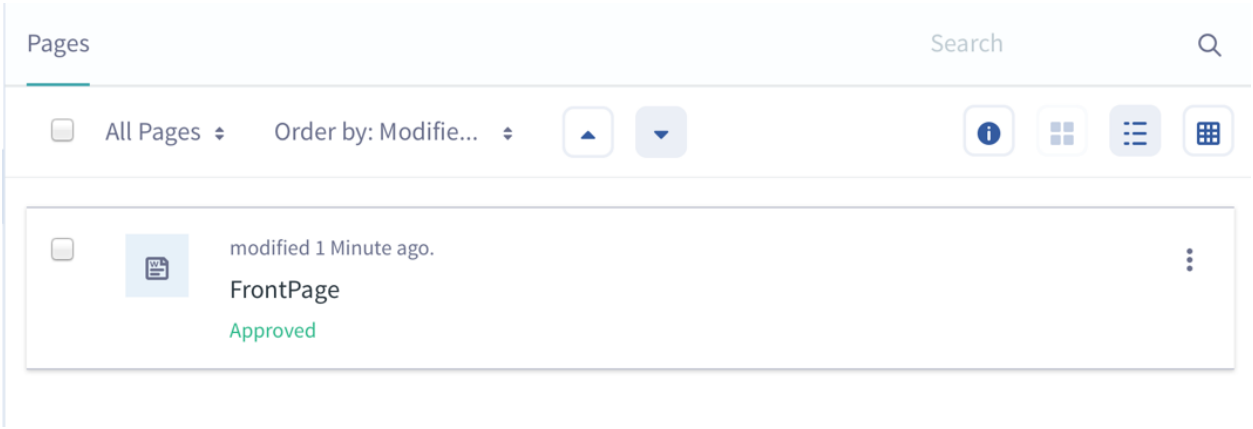


Figure 19.26: The wiki node's view in site administration has features that help you access and learn information about a wiki node's pages.

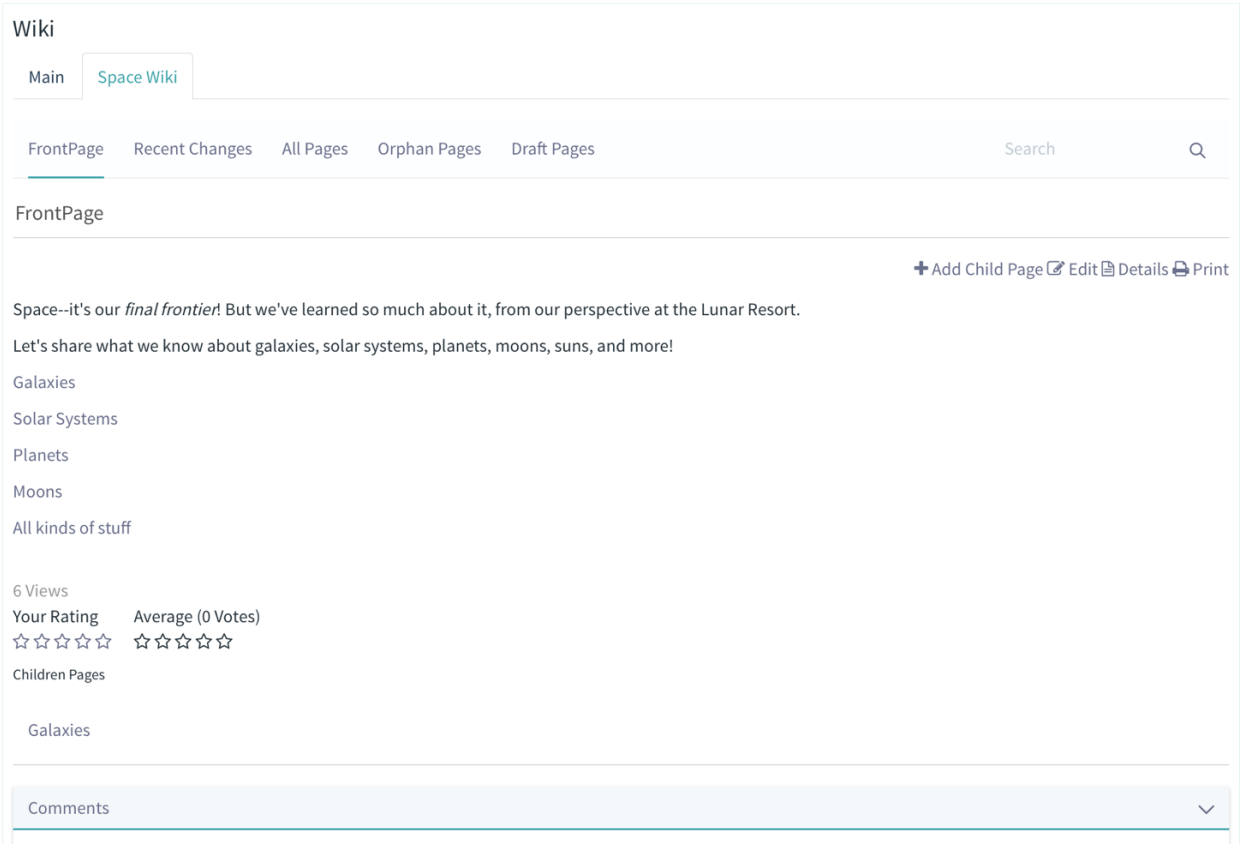


Figure 19.27: : Users can interact with your Wiki nodes when you add the Wiki application to a page.

Setup: lets you choose wikis to display and gives you several options for displaying them. The *Enable Related Assets*, *Enable Page Ratings*, *Enable Comments*, and *Enable Comment Ratings* check boxes let you enable or disable those features for the Wiki app. They give you the ability to set how you want users to interact with wiki documents: a little, a lot, or not at all. The *Display Template* drop-down option lets you choose the Application Display Template for the app. Below this, you can set which wikis (wiki nodes) are visible in the Wiki and which are hidden. You might host two wikis on a given site, exposing one to the public and keeping the other private for site members.

Permissions: enables you to grant or revoke privileges for roles to edit the Wiki application's preferences, configure it, access its permissions, view the Wiki application's contents, add display templates for it, or add the application to a page.

Communication: allows you to configure communication across portlets, using predefined public render parameters. From here you can modify six public render parameters: categoryId, nodeId, nodeName, resetCur, tag, and title. For each parameter you can:

- Ignore the values for this parameter that come from other portlets. For example, the wiki app can be used along with the tags navigation app. When a user clicks on a tag in the tags navigation app, the wiki shows a list of pages with that tag. In some cases an administrator may want the wiki app to always show the front page independently of any tag navigation done through other portlets. This can be achieved by checking the Ignore check box so that the values of the parameter coming from those other portlets are ignored.

- Read the value of a parameter from another app. This is an advanced but very powerful option that allows portlets to communicate without configuring it beforehand. For example, imagine that the wiki app is used to publish information about certain countries. Imagine further that a custom app that allows browsing countries for administrative reasons was written and placed on the same page. We could associate to this second app a public render parameter called *country* to designate the name of the country. Using this procedure, we can cause the wiki to show the information from the country being browsed through in the other app. You can do this here for the wiki by setting the value for the title parameter to be read from the country parameter of the other app.

Sharing: displays options you're likely to be familiar with such as the tab for sharing the application with websites, Facebook, and NetVibes.

Scope: allows you to select the Wiki application of a particular scope to display. You can select the site scoped or global scoped instance, or select or create an instance for the page. If the page doesn't already have an instance scoped to it, you can click on the *[page name] (Create New)* menu option to create a page-scoped Wiki application instance.

Once you set the application's configuration options the way you want them, click *Save*.

Along with the Configuration option, the Wiki application's options menu lists all of the standard application options. The options enable you to specify the app's look and feel, export or import app data, minimize or maximize the app in the browser, use Configuration Templates to store your current application setup or apply an existing archived setup to this application instance, or remove the app from the page.

The Wiki application displays links to all of the Wiki application instance's wiki nodes and provides links for navigating around the wiki. Simply click on a wiki node's name to begin browsing that node's wiki pages. The following navigation links are listed after links to the wiki nodes.

Recent Changes: takes you to a page which shows all of the recently updated pages.

All Pages: takes you to a flat, alphabetical list of all pages currently stored in the wiki.

Orphan Pages: takes you to a list of pages that have no links to them. This can happen if you take a page link out of a wiki page in an edit without realizing it's the only link to that page. This area allows you to

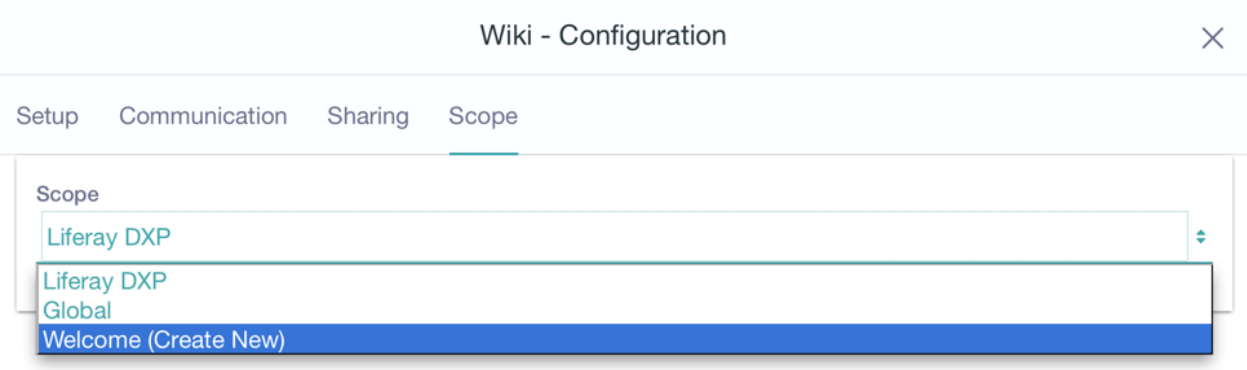


Figure 19.28: : Here the user has selected to create a new Wiki application instance scoped to the current page named *My Wiki*

review wiki pages that are orphaned in this way so that you can re-link to them or delete them from the wiki if they are no longer relevant.

Draft Pages: takes you to a list of pages which have not yet been published. Users can edit pages and save their changes as drafts. They can come back later to finish their page changes and publish them once they've been approved.

Search: allows you to a search for pages matching a term. If no matches are found for the term, a link displays to enable you to create a new wiki page named after that term.

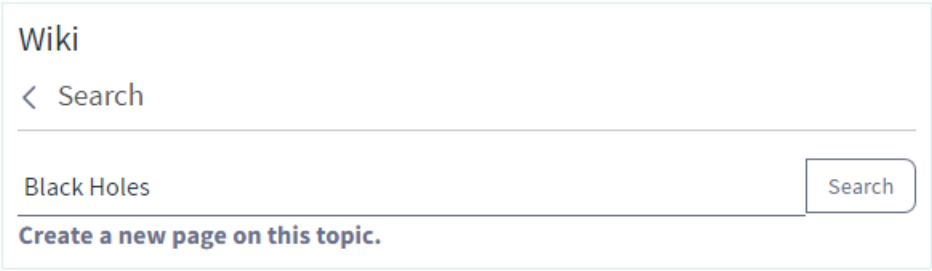


Figure 19.29: : If the wiki node doesn't have a wiki page matching the topic you're looking for, it gives you the option to create a wiki page named for the topic.

The current wiki page's content shows in the application's main viewing area. Several features display below the wiki page content, depending on which features are enabled for the application. The *+ Add Child Page* link lets you add a wiki page as a child of the current wiki page. A view counter displays the wiki page's view count. Ratings and Comments controls display if they're enabled.

The Edit, Details, and Print icons show above each wiki page's content. Clicking the Edit icon brings up the edit screen. Clicking the Print icon brings up the browser's page printing window. Between these two options is the *Details* option, which we'll explore next.

Page Details

When viewing a page, you can view its details by clicking the *Details* icon above the page content. Several tabs appear, to give you access to several categories of information about the page.

Details The Details tab shows various statistics about the page, and also allows you to perform some actions on the page.

Title: displays the page title.

Format: displays the format for the page – either Creole, HTML, MediaWiki, or plain text.

Latest Version: displays the latest version of the page. The wiki portlet automatically keeps track of page versions whenever a page has been edited.

Created By: identifies the user who created the page.

Last Changed By: identifies the user who last modified the page.

Attachments: displays the number of attachments to the page.

RSS Subscription: displays an icon that opens a new page where you can subscribe to an RSS feed using Live Bookmarks, Yahoo, Microsoft Outlook, or an application you can choose from your machine.

Email Subscription: contains links allowing you to subscribe to or unsubscribe from modifications notifications for the page and the entire wiki node.

Advanced Actions: displays icons allowing you to modify the permissions on the page, make a copy of the page, move (rename) the page, or move the page to the recycle bin.

History The History tab lets you access the page's activities and versions. Its *Activities* tab lists actions performed on the page. Each activity has an icon that represents the type of action, the name of the user, the actions description, date, and an options icon to revert the action. For example, if user Jane Doe attached file `sunset.jpg` to the page, then the activity would show an Attachment icon ( ), this text *Jane Doe added the attachment `sunset.jpg`.*, the action date, and an options icon to remove the attachment.

The *Versions* tab lists all of the versions of the wiki page since it was created. You can revert a page back to a previous state by clicking the *Revert* icon ( ). You can also compare the differences between versions by selecting two versions and then clicking the *Compare Versions* button.

Incoming/Outgoing Links The next two tabs are list incoming and outgoing links. These are wiki links to and from the page. You can use this tab to examine how this page links to other pages and how other pages link back to this page.

Attachments The *Attachments* tab lists the name and size of each file attached to the page. You can attach any file to the wiki. Image files are the most common type of file attached to a page. Referencing them using the proper WikiCreole syntax renders the image inline, which is a nice way to include illustrations in your wiki documents.

As you've seen, Liferay's Wiki application is chock-full of features that help you create, maintain, and consume wiki pages. In addition to the Wiki application there are several other applications that supplement it. You'll learn about them next.

Supplemental Wiki Applications

The applications that accompany the Wiki application help you display particular wiki nodes and navigate them. To use them on a page, click the *Add* icon ( ), navigate to *Applications* → *Wiki*, and either click *Add* next to the application you want or drag it onto the page.

Let's first explore the Wiki Display.

Wiki Display The Wiki Display application enables you to focus user attention on one wiki node. Click on the app's Options icon and select *Configuration*. In the Configuration screen, select the *Setup* tab. Set the

Node you want to display and click *Save*; then select a *Page* in that node that you want to display. This page provides the gateway into the wiki node.

Figure 19.30: : The Wiki Display lets users explore a wiki node, starting from one of its pages—the front page, typically.

The configuration options and user interface for the Wiki Display are almost identical to that of the Wiki application.

To help users navigate the wikis in a site page's Wiki Display or Wiki applications, you can use the Tree Menu app.

Tree Menu The Tree Menu application displays a wiki's page hierarchy as a tree. It lets you navigate all of a wiki's pages. Much like Wiki Display setup, you configure the Tree Menu app to focus on a wiki node. In addition, you can configure how deep to allow users to navigate into the page hierarchy. You can set the *Depth* to a value from 1 to 5, or select *All* to allow navigation to all of the wiki node's pages.

In the Tree Menu, folder icons represent parent wiki pages and document icons represent child wiki pages at the end of the nodes. When you click on a parent wiki page or child wiki page icon, the Wiki application or Wiki Display on the site page displays the respective wiki page.

There's even a Page Menu application that presents a wiki page's outgoing links.

Page Menu The Page Menu app lets you display a wiki page's outgoing links. It answers the question, "What wiki pages can I access from this page?" As with the Tree Menu and Wiki Display applications, you specify a wiki node and wiki page on which the Page Menu application operates.

When you click on a Page Menu link, the site page's Wiki or Wiki Display application displays the wiki page associated with the link.

Summary

Liferay DXP's Wiki application is another full-featured Liferay application with all of the features and sub-applications you expect from a state of the art wiki. It allows you to create pages of information quickly and collaboratively. The Wiki gives administrators flexibility to reposition wiki pages to their proper place in a wiki hierarchy. And it allows users to author rich content to share with others.

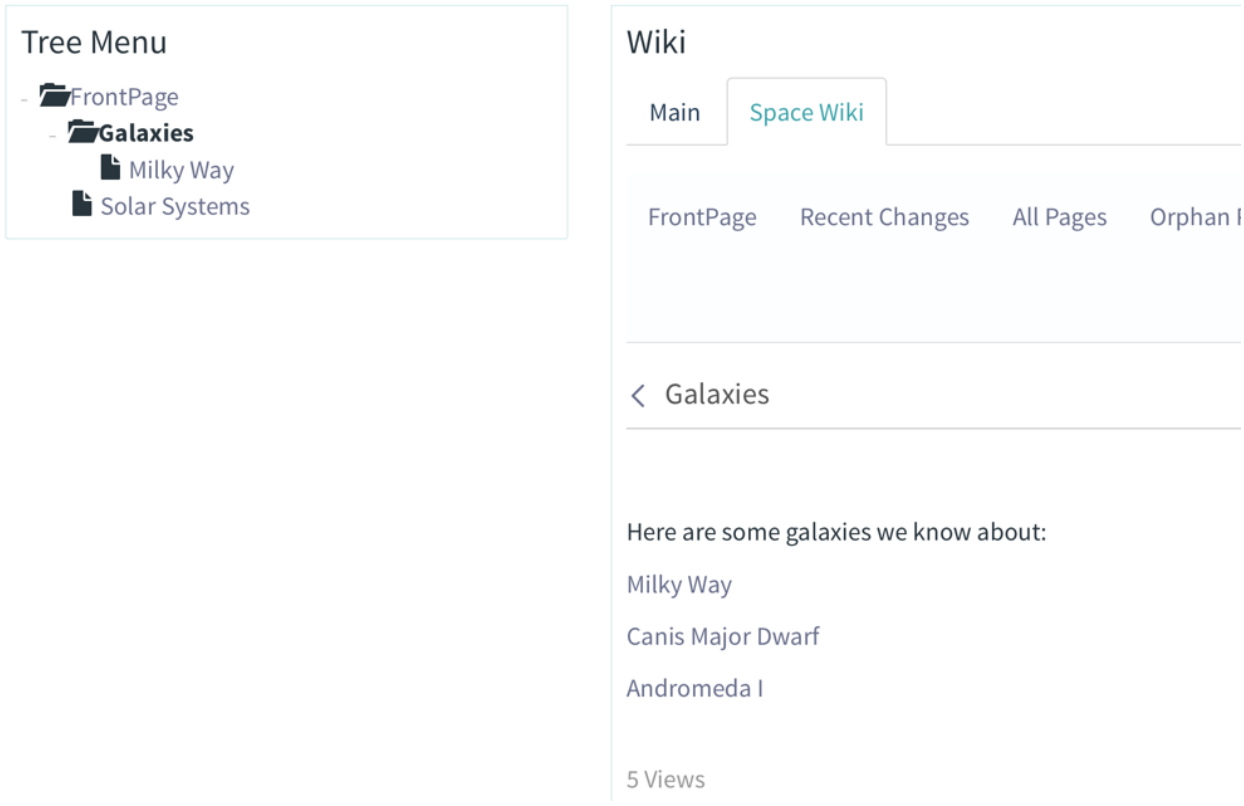


Figure 19.31: : When you select a element in the Tree Menu application, the respective wiki page shows in your site page's Wiki application or Wiki Display.

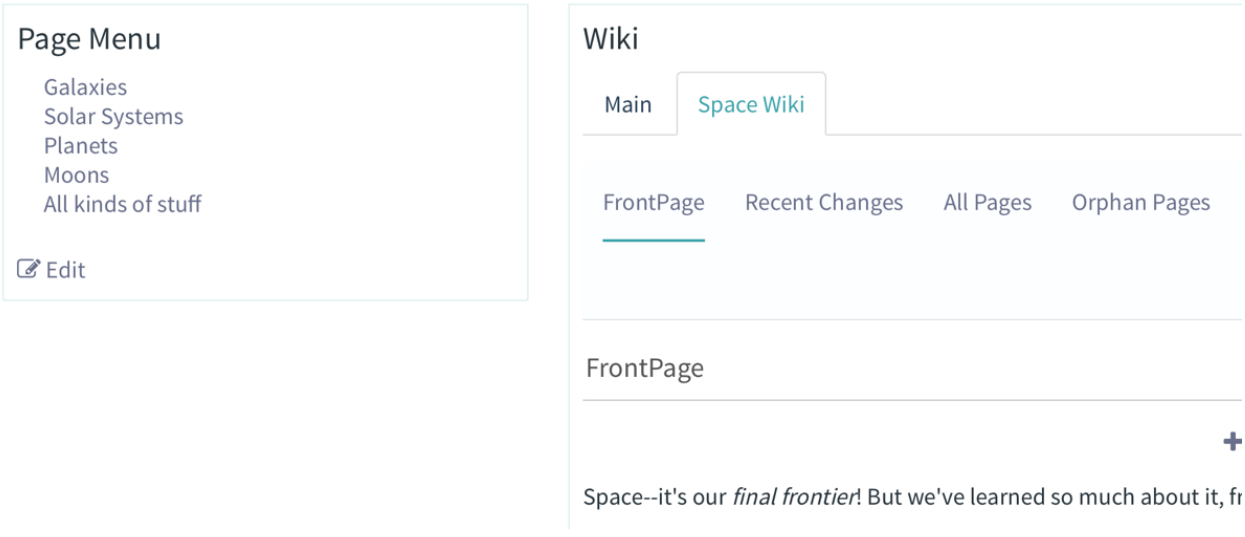


Figure 19.32: : The Page Menu application displays all of a wiki page's outgoing links to other wiki pages.

Next we'll learn how to inform and alert users with the Alerts and Notifications applications.

19.5 Sending Alerts and Announcements

Suppose you're running a Liferay instance with lots of users. Maybe you have thousands or tens of thousands of users (or more!). Wouldn't it be nice to have an easy way to periodically communicate important information to them? And wouldn't it be even nicer to have easy ways to periodically send information to specific groups of users? For example, maybe you'd like to send a reminder about a new policy to all of your Liferay instance's site administrators. Or maybe you'd like to send a reminder to all the members of a certain site to submit an answer to this week's poll question. Liferay provides two apps that meet this need: the Alerts app and the Announcements app.

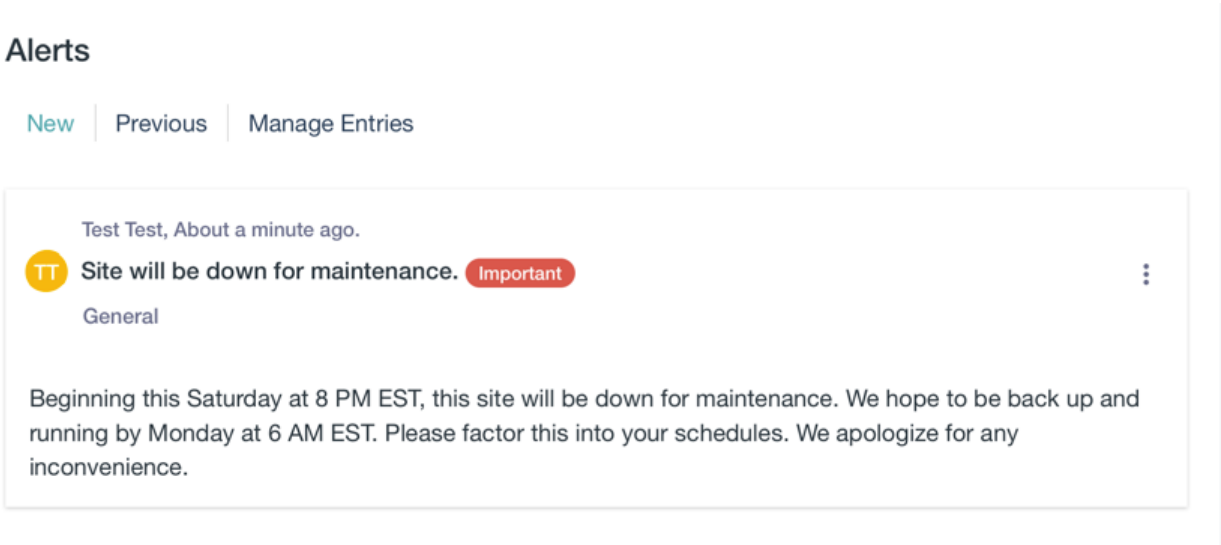


Figure 19.33: The Alerts app provides administrators with an easy way to communicate important information to appropriate groups of users.

The Alerts and Announcements apps let you broadcast important information to different groups of users. Each alert or announcement can be created in a specific scope so that you can manage which announcements are sent to which users. Furthermore, each user can configure how they'd like to receive announcements: click on *[User Name]* → *My Account* → *Account Settings* from the Menu and then on *Announcements* in the Miscellaneous category. This brings up a menu for customizing the delivery options for alerts and announcements. You can select a different configuration for each type of alert or announcement: General, News, or Test. For each type, you can enable delivery by email and SMS (text message). Note that the *Website* delivery option is checked and grayed out for each alert type. This means that each alert and announcement is always viewable in its respective app on a site.

You can also create roles in Liferay for users to make general announcements. For instance, if you'd like an employee in your site to have strict control over what is announced, you can assign them to an Announcements role. To create a simple Announcements role, navigate to *Control Panel* → *Users* → *Roles*, click the *Add* button ( ), and select *Regular Role*. Name your role *Announcements*, give it a title and description, and click *Save*. Then click the *Actions* button ( ) next to your new role and select *Define Permissions*. For this particular role, you'll need to grant two permissions:

- *Control Panel* → *General Permissions* → *Add General Announcements*

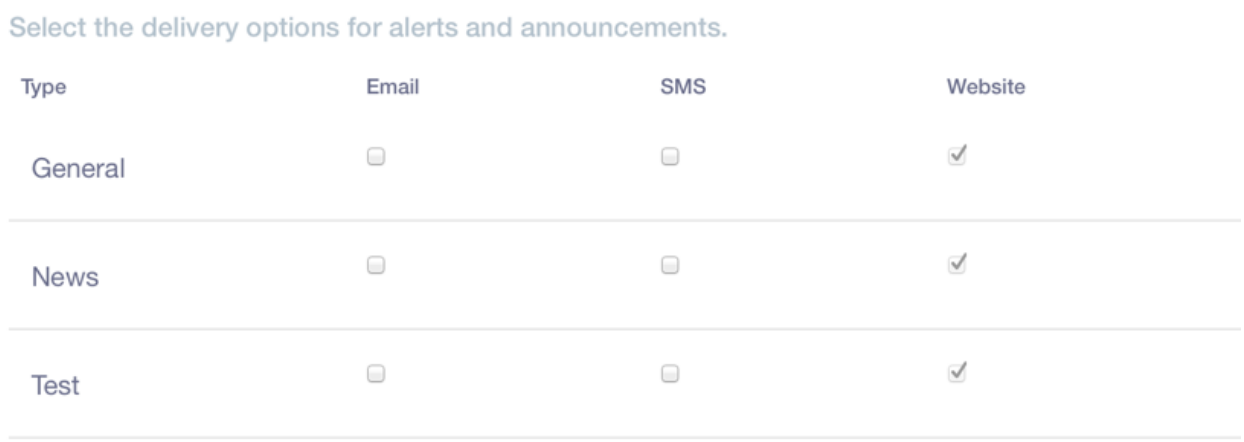


Figure 19.34: Each user can choose how they receive alerts and announcements.

- *Site Administration → Applications → Announcements → Add Entry*

Click *Save* after selecting each permission. Now you have a simple Announcements role that can add an entry to the Announcements app and add general announcements to your site! Of course, you can add additional permissions to this role if you prefer.

Now you'll learn how to create announcements in the Announcements app. When you view the Announcements app as an administrator, you'll see three tabs: New, Previous, and Manage Entries. The New tab displays non-expired announcements, and announcements you haven't marked as read. Once an announcement expires or you mark it as read, it moves to the Previous tab. Administrators can use the Manage Entries tab to create new announcements. Non-administrators only see the New and Previous tabs. To add a new announcement, click *Manage Entries* and then select a *Distribution Scope*. The distribution scope determines the group of users to which your announcement is sent. Announcements created in the General distribution scope are sent to everyone. If the distribution scope is set a particular site or role, only members of that site or role receive the announcements.

Once you've selected a distribution scope, you'll see a list of that scope's existing announcements. Click an announcement's *Actions* button to edit or delete it. Click *Add Entry* to create a new announcement. In addition to the Distribution Scope, you can provide the following information:

Title: The announcement's title. When the app displays the announcement, the full title always appears in bold. If you enter a URL in the URL field, the title serves as a link to the specified site.

URL: The URL is optional. If entered, it must be valid and begin with *http://* or *https://*. For example, an announcement about a news story could include a link to the news article.

Content: The announcement's body text. You can use the familiar CK editor to write this. Use the *Source* button to switch between the editor and HTML views.

Type: The announcement's type. This can be *General*, *News*, or *Test*. As mentioned earlier, each user can specify a different delivery mechanism for each type of announcement. For example, a user could choose to receive General announcements via email and text, News announcements via email only, and no special delivery mechanism for Test announcements. Each user can customize their delivery options by navigating to *[User Name] → My Account → Account Settings* from the Menu, and then selecting *Announcements* in the Miscellaneous category.

Priority: The announcement's priority. This can be *Normal* or *Important*.

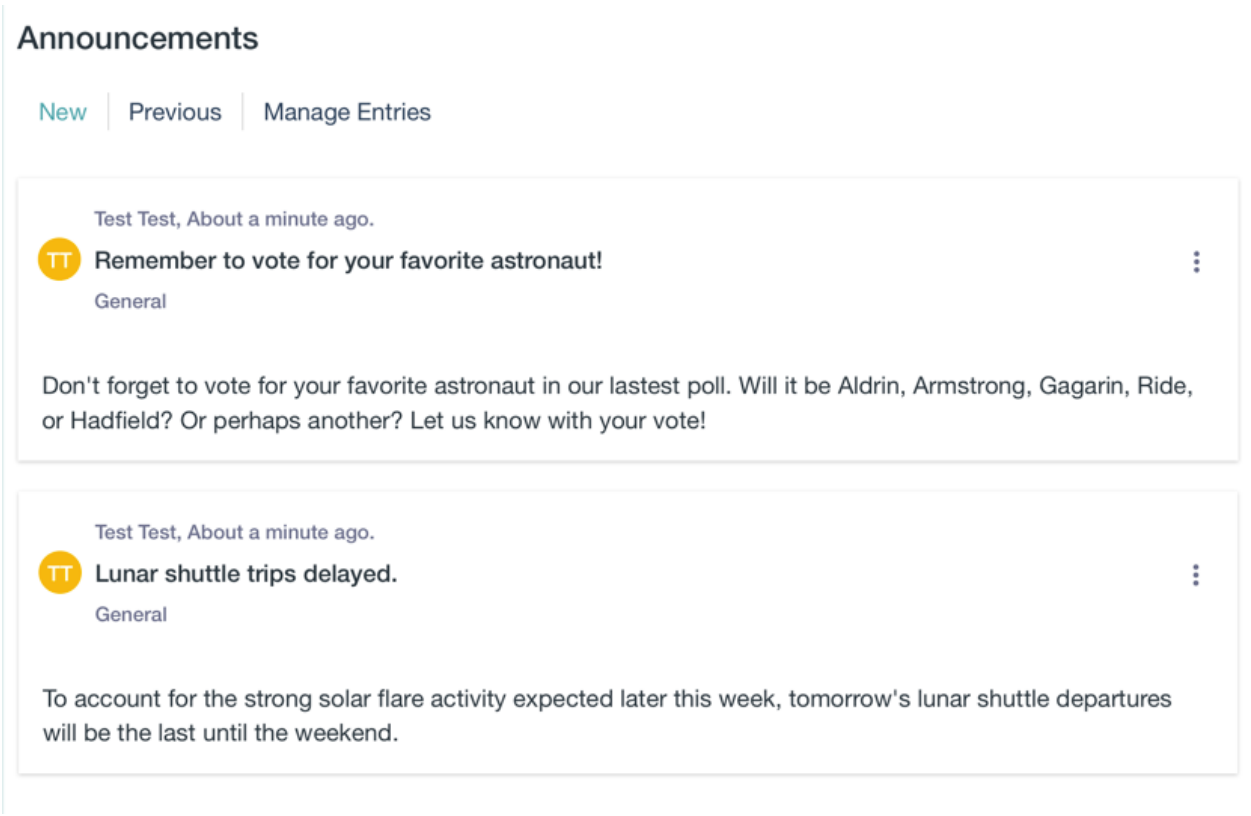


Figure 19.35: The Announcements app displays a list of announcements to regular users. It also provides administrators with an interface for managing and adding entries.

Display Date: The announcement's display date. This determines when the announcement is sent to users and appears in the Announcements app. By default, the *Display Immediately* box is checked. This sets the display date equal to the creation date. Uncheck this box to enter a custom display date. For example, administrators can create announcements for display on a later date. This date can be days, weeks, months, or years in the future. Once the *Display Immediately* box is unchecked, clicking the Display Date field opens the date-picker widget.

Expiration Date: The date and time the announcement expires. Once an announcement expires, the Announcements app displays it in the Previous tab. Clicking the Expiration Date field opens the date-picker widget.

The Alerts app works the same way as the Announcements app. Alerts is a separate app because it's designed to be used only for high priority messages or those that require users to take some action. The Alerts app only displays alerts, and the Announcements app only displays announcements. To draw attention to its alerts, the Alerts app displays a red *Important* tag next to each. To separate important alerts from more mundane announcements, site administrators can place the Alerts and Announcements apps on different pages.

19.6 Managing Notifications and Requests

If you subscribed to a blog or message board, or if someone sent you a private message, invitation, event reminder, or mentioned you in a post, you received a notification or request.

The notifications icon on your profile image shows your number of unread notifications or requests.

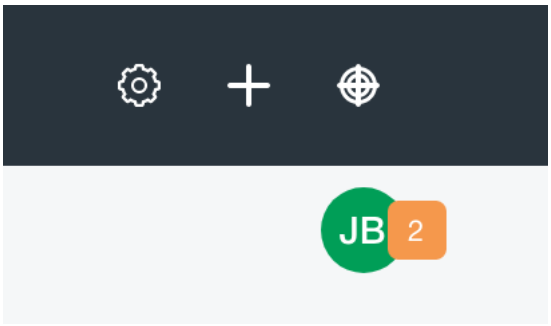


Figure 19.36: The number of notifications and requests are displayed above your profile image

To access notifications and requests, you can click the notifications icon on your user profile image or open the Control Menu and select *My Account* → *Notifications* under the user menu. The *Notifications List* section is selected by default. This is where all your notifications are displayed. Click the *Requests List* link in the Navbar to view and manage your requests.

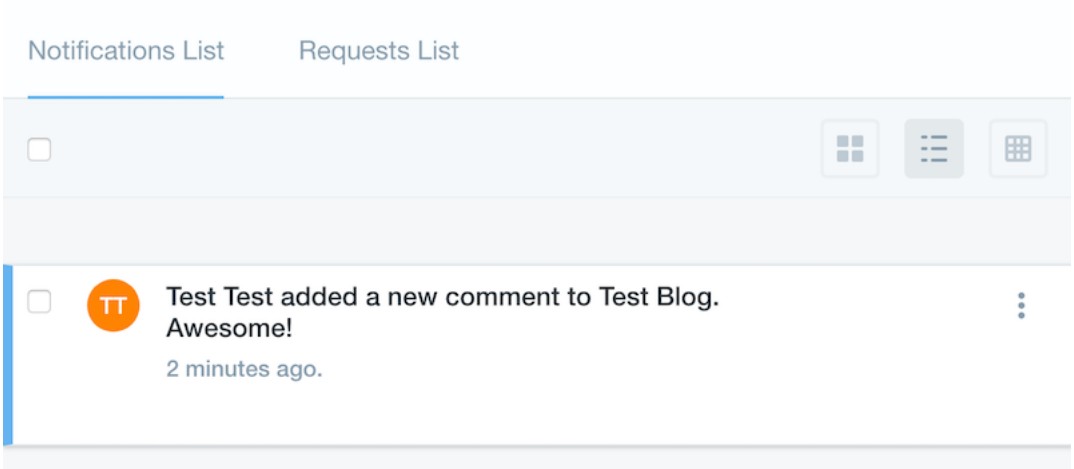


Figure 19.37: The *Notifications List* section displays all your notifications in a paginated list.

Managing Notifications

Notifications can pile up after some time, especially if you were away for a few days. Since Liferay DXP 7.0 Fix Pack 30 and Liferay Portal 7.0 CE GA5, the Management Bar gives you several ways to filter and sort your notifications.

The following filters are available for notifications:

All: The default option. Displays both read and unread notifications.

Unread: displays notifications that haven't been marked as read. Unread notifications are indicated with a blue border on the left-hand side of the notification.

Read: displays notifications that have been marked as read

You can also sort notifications by *Date*. By default notifications are listed by date in descending order. Click the up arrow icon in the management bar to sort notifications by ascending order. The down arrow icon reverses this.

To mark notifications, select *Mark as read* or *Mark as unread* from the actions menu for the notification. The *Delete* action permanently deletes the notification.

Managing Multiple Notifications You can also manage multiple notifications at once. Select the checkbox next to notifications you want to manage and choose an option from the Management Bar. Select the checkbox above the notifications list to select all notifications on the current page. A blue border on the bottom marks selected notifications. Selected notifications show three actions: an open envelope button ( ) (mark as read); a closed envelope button ( ) (mark as unread); and an x button ( ) (delete).

Selecting the *Mark all notifications as read* action in the app's actions menu marks them as read.

Note: This feature is only available if at least one notification is unread. Clicking it marks all notifications as read, selected or not.

This action can not be undone. You must manually paginate through all the notifications and either bulk mark them as unread for the page, or individually mark the notifications as unread if you wish to change the state.

Managing Requests

When you get a request, it appears in the *Requests List* section. In its Action menu, you can click *Confirm* to accept, *Ignore* to decline, or *Delete* to remove the request.

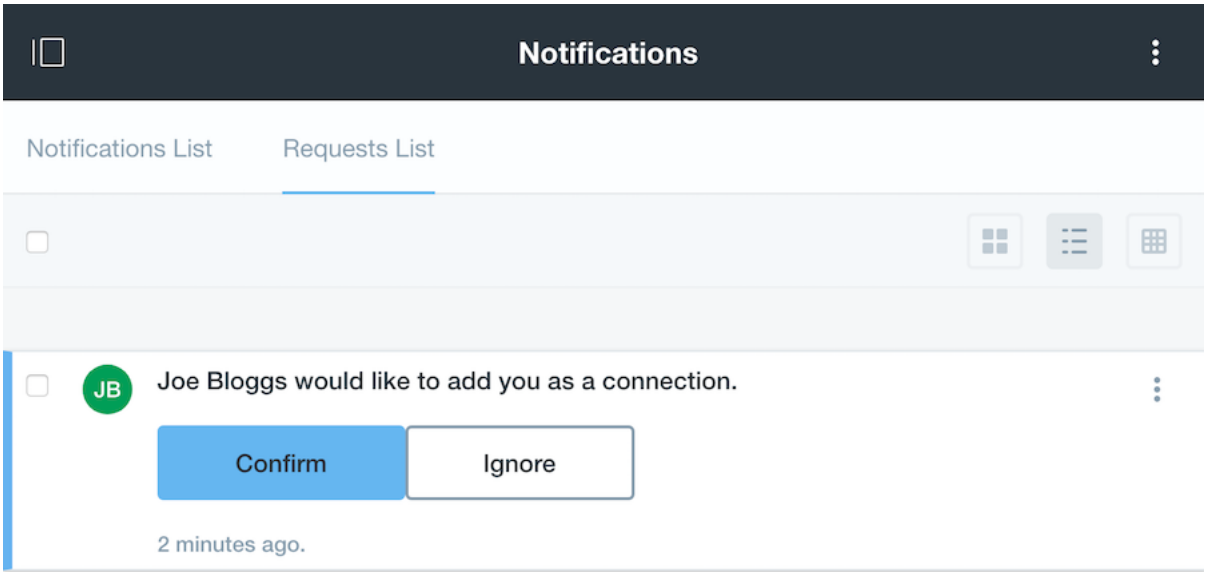


Figure 19.38: You can confirm, ignore, or delete requests.

19.7 Informing Users with the Knowledge Base

Liferay's Knowledge Base app lets you create and organize articles in a site. For example, it can be used to display professional product documentation or form complete books or guides. It even lets you import article

source files written in Markdown. It's workflow-enabled, so you can require articles to be approved before they are published. Additionally, the Knowledge Base app lets administrators create article templates that help users follow a common outline. With this application, you can create and manage your own knowledge base, full of information relevant to your users.

Note: Staging is supported in both Knowledge Base content and portlet configurations since Liferay DXP 7.0 Fix Pack 14 and Liferay Portal CE GA4.

Here are the key features of the Knowledge Base:

- Its navigation is built into the Knowledge Base Display.
- It has a suggestions interface for article feedback.
- It stores articles in folders.
- It contains metadata fields for friendly URL, source URL, categorization, and related assets.
- The button (

) can be enabled to take readers to an article's source repository location (if you choose to use it that way).
- It imports article Markdown source files to create and update articles.

The Knowledge Base has several portlets you can add to site pages:

- Knowledge Base Article
- Knowledge Base Display
- Knowledge Base Search
- Knowledge Base Section

Each of these is covered below. First you'll learn how to create and manage articles.

Creating and Managing Articles

Site Administration contains everything that administrators and authors need to create, update, and manage articles in the Knowledge Base.

To navigate to the Knowledge Base, go to Site Administration → *Content* → *Knowledge Base*. It has two tabs: one for creating and managing articles and article folders (Articles) and one for managing user-submitted feedback for articles (Suggestions).

Note: To access Knowledge Base in Site Administration, a role must have the permission *Knowledge Base → Access in Site Administration*. To add or act on articles, folders, or suggestions, the site administrator must grant the appropriate permissions using the Permissions window in Knowledge Base.

To create an article follow these steps:

1. Select the *Articles* tab and choose *Basic Article* or the name of an available template in the Add button () menu.

   This brings up the New Article form.

2. Enter a title for the article.

   A URL-safe version of the title you provide is added to the end of the article's friendly URL. You can manage the friendly URL in the *Friendly URL* field underneath the *Configuration* section heading.

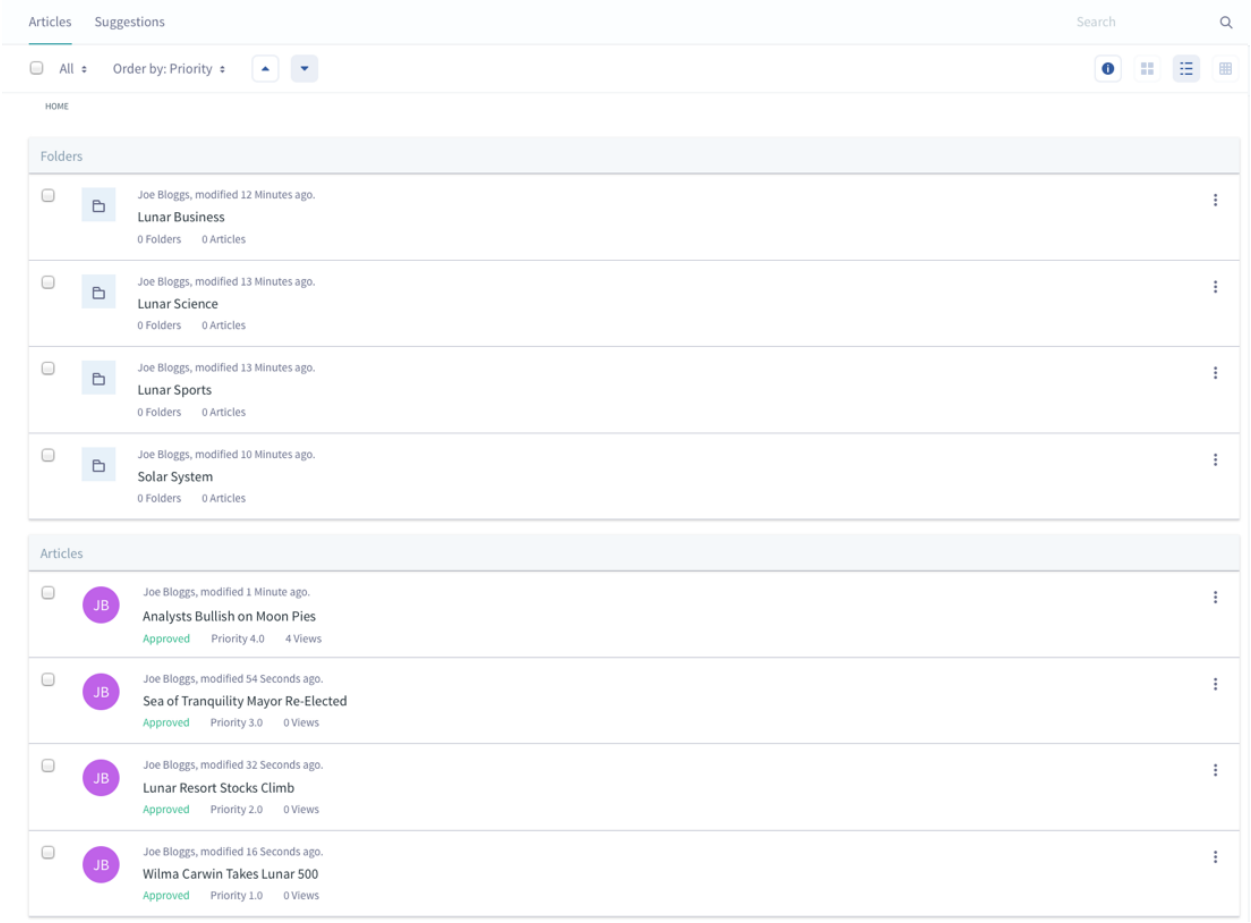


Figure 19.39: You can manage Knowledge Base articles, folders, and suggestions from the Knowledge Base in Site Administration.

3. Use the WYSIWYG editor to create the article's content.

   Click the *Source* button in the editor to view the article's HTML source or write in HTML yourself. You can also add attachments and tags, specify related assets, and set permissions for the article in the sections below the editor. By default, View permission is granted to the Guest role, meaning anyone can view your article.

4. Click *Publish* to submit the article for publication or click *Save as Draft* to continue working on it later.

   Note: If you've enabled workflow for the Knowledge Base, your article must be approved before publication.

Once the article is saved, it is converted automatically to HTML for the Knowledge Base. Articles are listed in a table at the bottom of the Articles tab.

Each article also has a *priority* value that determines its position in the Knowledge Base Display app's (more on this app in a moment) navigation. Each article's priority value is listed underneath the article's title. The Knowledge Base Display app's navigation arranges articles in ascending priority. Priority 1 is the highest priority; think "Priority 1 distress call!" The higher an article's priority, the higher it appears in the navigation. Articles are assigned the next lowest priority by default. This behavior can be changed via System Settings.

To assign articles a new priority value, follow these steps:

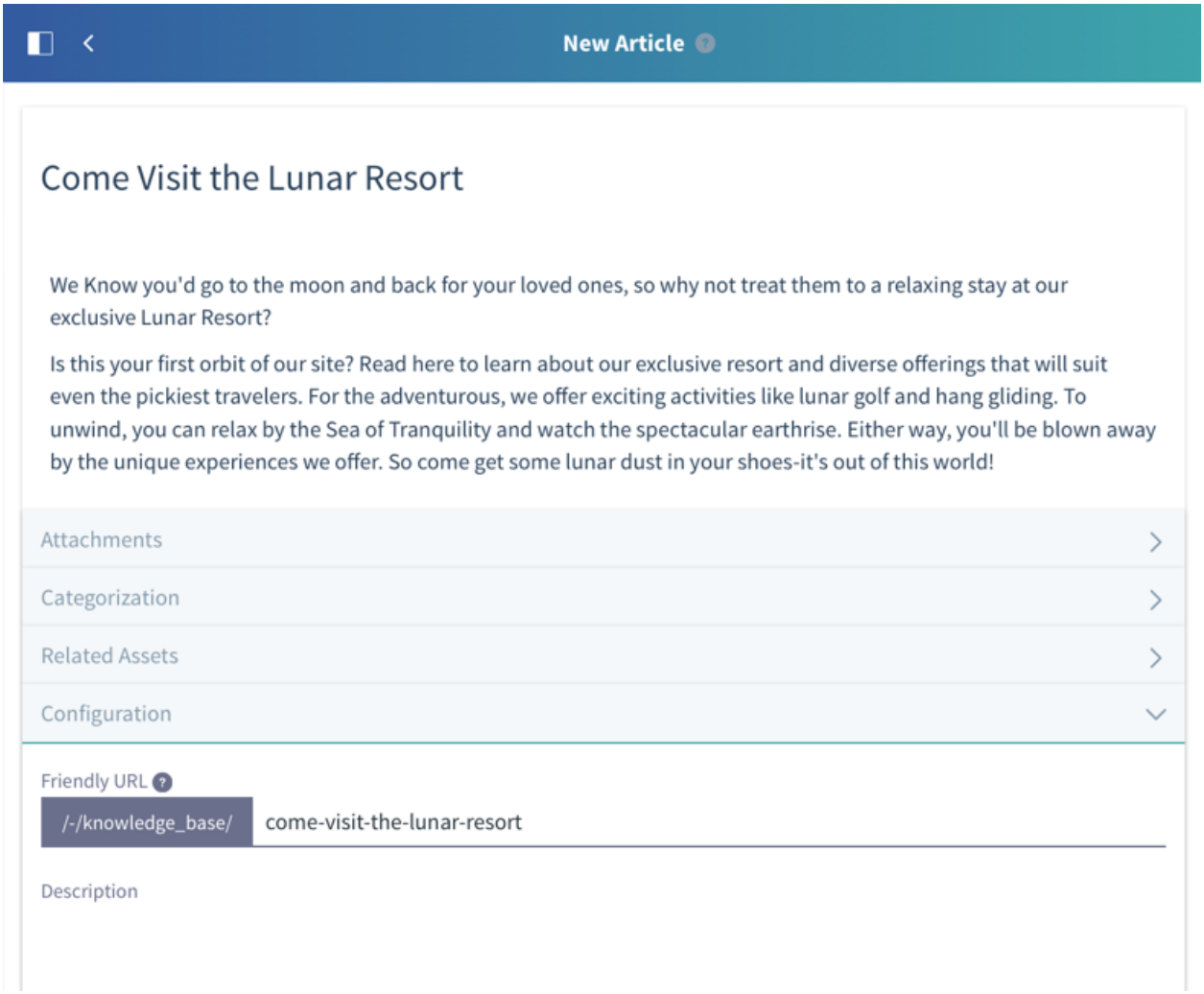


Figure 19.40: You can create and modify a Knowledge Base article's content using the WYSIWYG editor.

1. Select *Move* from the Actions menu next to the article.
2. Enter a new priority value for the article.
3. Click *Move* to apply the new priority.

You can also organize articles with folders. Follow these steps to create a folder:

1. Open the Add button menu and select *Folder*.
2. Enter a name and an optional description.

   By default, anyone can view the folder. You can manage this setting along with the other permissions through the *Permissions* heading.
3. Click *Save*.

   Folders are listed in a table near the top of the Articles tab.

The text immediately below the filters and *Order by* selector at the top of the app shows your position in the folder hierarchy. Click on a folder's name in the hierarchy to navigate to it. You can also move articles into folders and create child articles. Knowledge Base also supports nested folders.

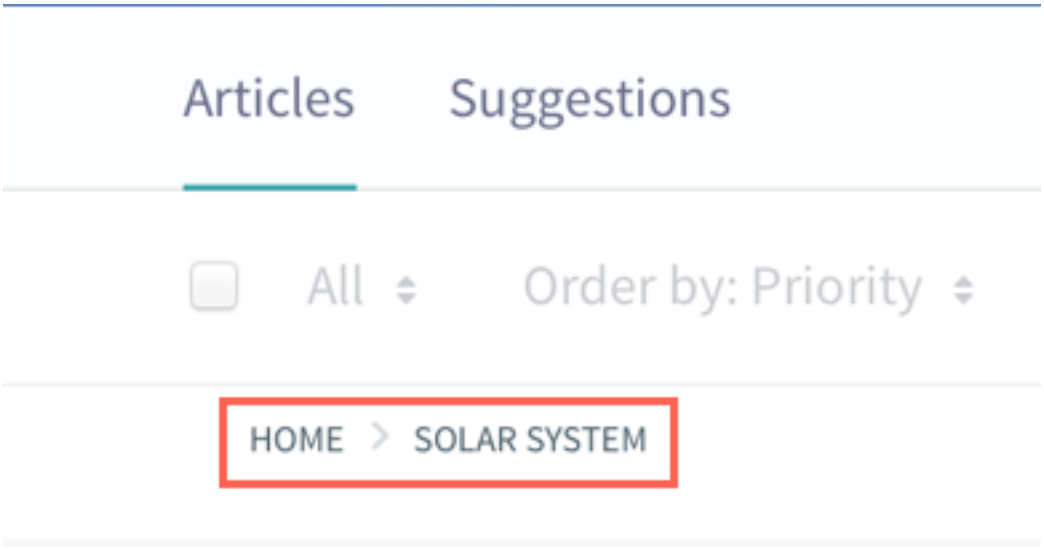


Figure 19.41: This screenshot uses a red box to highlight the text that indicates the current position in the folder hierarchy.

You can perform the following actions on folders:

- **Edit:** change the folder's name and description.
- **Move:** relocate the folder under a new parent folder or update its priority.
- **Delete:** remove the folder and its articles from the knowledge base.
- **Permissions:** grant or revoke the following permissions: add an article to the folder, add a sub-folder to the folder, delete the folder, move the folder, set permissions on the folder, edit (update) the folder, and view the folder.

You can also delete multiple articles or folders at once:
Select the checkbox for each item that you want to delete and click the *X* button that appears at the top of the app to delete it.
This Actions menu also lets you check the info for selected items. Note that this menu only appears when one or more article or folder checkboxes are checked.
The (+) menu provides the following options for articles:

- Add a Basic Article
- Add an article based on a template
- Add a folder
- Import articles from a ZIP file

Note: To import articles, your role must be granted the *Knowledge Base → Resource Permissions: Import Articles* permission.

The Options menu (⋮) has these functions:

- Manage templates
- subscribe
- edit permissions
- import/export configuration settings
- configure email options

The *Home Folder Permissions* option lets you define detailed permissions for the Knowledge Base app. You can choose the roles that can perform the following tasks:

- Add/delete articles, folders, and templates
- Change the Knowledge Base app's permissions
- Subscribe to articles
- View templates and suggestions

Next, you'll learn how to manage article templates.

Creating and Managing Templates Templates give users a starting point to work from. For example, you can create templates that contain default headers or other content for articles. Templates help foster consistent formatting and content organization for articles.

To create a new template follow these steps:

1. Click the *Templates* button in the Options menu ( ) at the top of the app.

   This brings up the Templates page, where you can manage existing templates.

2. Click the Add button at the bottom of the screen to add a template.

3. Fill in the New Template form and click *Publish*.

From the Templates page, the following actions are available in each template's Actions button:

- **View:** display the template. From here, you can print the template, use it to create an article, edit it, modify its permissions, or delete it.

- **Edit:** change the template's title and content.

- **Permissions:** configure the permissions on the template. You can choose whether a role can change permissions, update, view, or delete the template.

- **Delete:** remove the template from the Knowledge Base.

Knowledge Base in Site Administration also lets you handle user feedback on articles. You'll learn about this next.

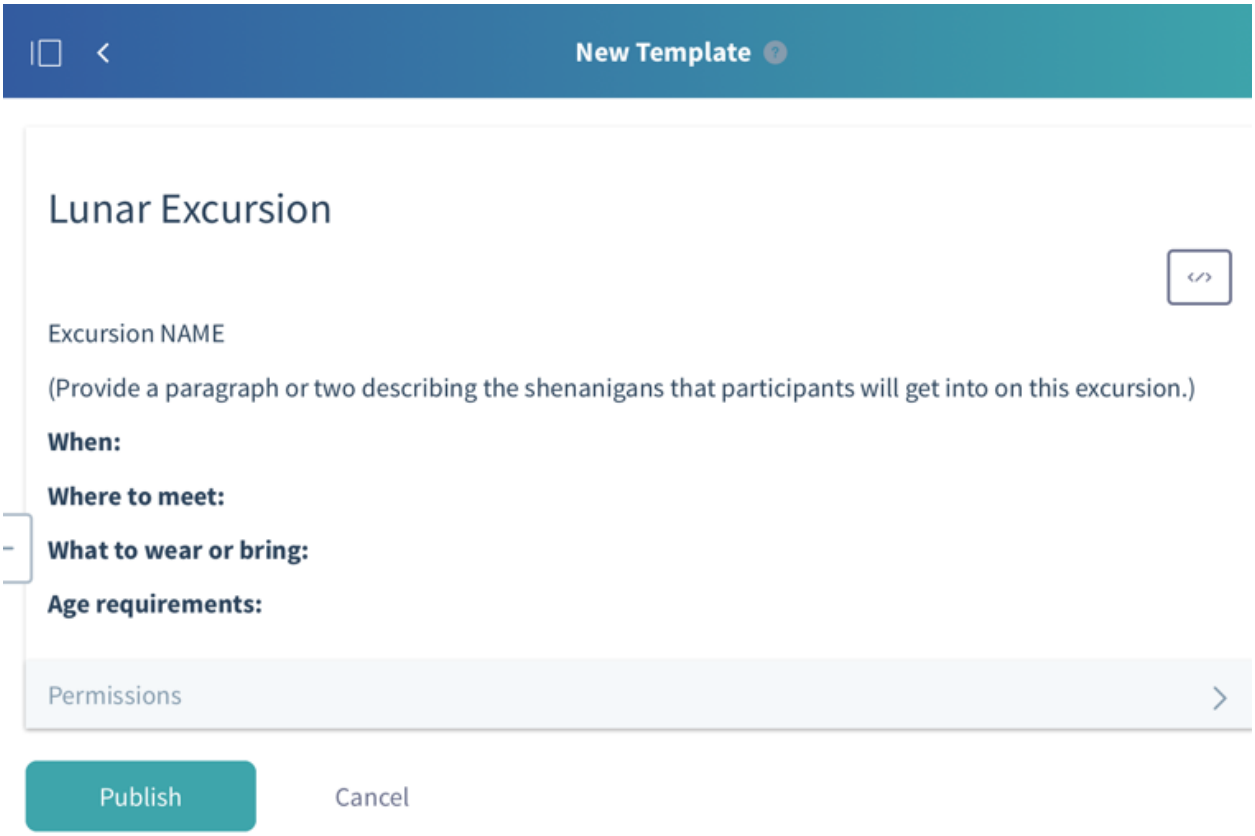


Figure 19.42: As an administrator, you can add a new template to your knowledge base from Knowledge Base in Site Administration.

Responding to User Feedback Knowledge Base's Suggestions tab shows user feedback on articles and lets you mark progress on addressing the feedback.

Each suggestion provides the link to the associated article, the user's feedback, the user's name, the feedback's time stamp, and the status on addressing the suggestion. You can move the entry between *New*, *In Progress*, and *Resolve* states.

Note: To view article suggestions, your role must be granted the *Knowledge Base → Knowledge Base: View Suggestions* permission.

To move suggestions between the *New*, *In Progress*, and *Resolve* states, your role must be granted the *Knowledge Base → Knowledge Base Article: Update* permission. Roles assigned this permission can also view and update the state of article suggestions from any of the other Knowledge Base apps.

When you move the suggestion to a different state, an email is sent notifying the user of the change. To view and configure the templates for the automated emails, click the *Configuration* button in the Options menu ( ).

So far, you've learned how to create, edit, and organize articles. You've also seen how the suggestions feature lets you and your users collaborate on your content. Next, you'll learn how the other Knowledge Base apps let you display, navigate, aggregate, and view articles.

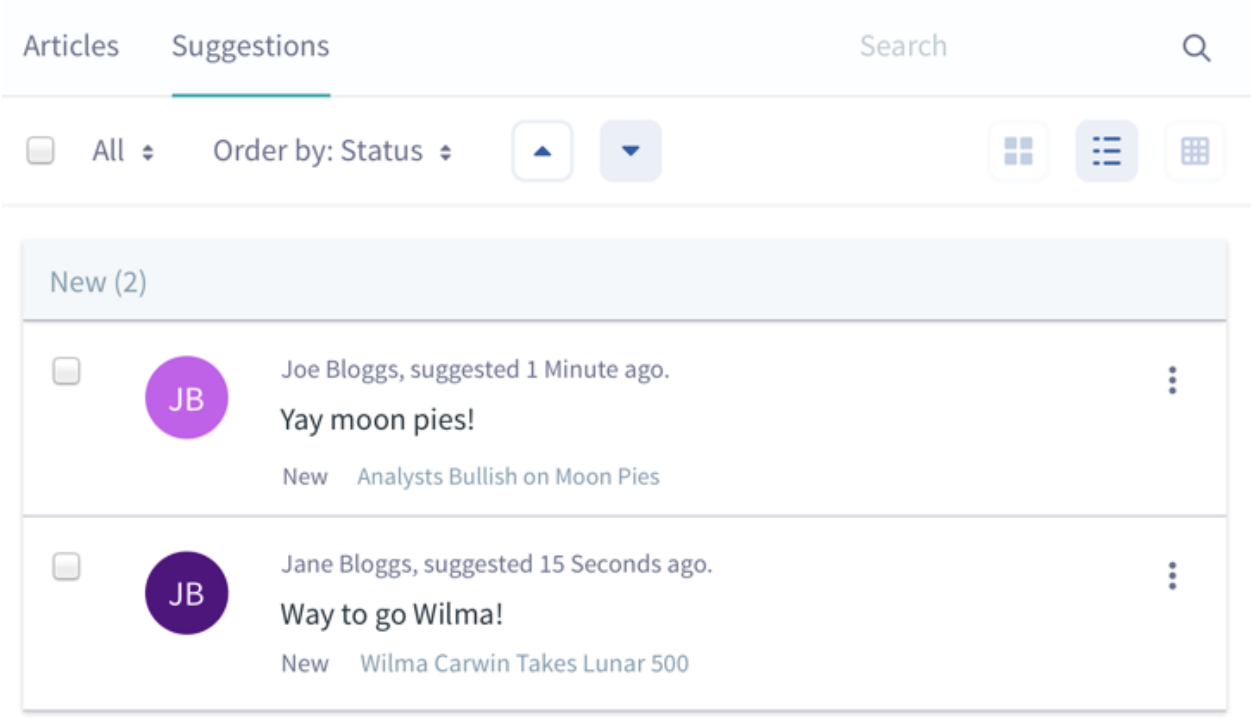


Figure 19.43: The Suggestions tab in Knowledge Base helps you monitor article feedback and mark progress in addressing feedback.

Knowledge Base Display

You can use the Knowledge Base Display app to display your published articles. Knowledge Base Display is flexible. You can customize how it displays articles, as well as which ones it displays. Add the Knowledge Base Display to the page where you're publishing your articles.

To display an article or folder of articles, follow these steps:

1. Mouse over the app, click the *Options* icon ( ) in the top right corner, and select *Configuration*.
2. Click *Select* in the *Article or Folder* field to bring up the available folders and articles.
3. Click *Choose* next to the article or folder of articles you want to display, and *Save*.

   You can also click the *Choose This Folder* button at the top of the Select Entry dialog to select the current folder.

Knowledge Base Display also provides these common configuration options:

- look and feel
- export/import
- permissions
- scope
- sharing

For more information on this, see the section on configuring Liferay applications in Web Experience Management.

The Knowledge Base Display's navigation menu and display options make it the perfect candidate for a full page app. If you display a folder of articles, the navigation on the left side of the app displays links to all the folder's articles. The viewing area on the right side of the app displays the folder's leading article (the "priority one" article). Click an article in the navigation to display it in the viewing area. The currently displayed article's link appears in bold in the navigation. You can also move between articles by clicking the links at the bottom of the app.

Figure 19.44: Knowledge Base Display's navigation and viewing area facilitates viewing article hierarchies.

Knowledge Base Display can also show article hierarchies. Viewing an article that has child articles expands the navigation tree to show links to the child articles. Any expanded nodes collapse when you view a different top level article.

The links at the top of the app allow users to perform the following actions on an article:

- Subscribe to an RSS feed of the Knowledge Base
- subscribe to the current article
- view the current article's history
- Print the current article

Administrators have access to an additional set of links at the top of the app that allows them to perform the following actions:

- Edit the article
- Add a child article
- Assign permissions to the article
- Move the article
- Delete the article

Below the article's content is the rating interface, showing thumbs up and thumbs down icons. Users can also submit suggestions or comments below the article in the *Do you have any suggestions?* text box. Administrators can view the suggestions and mark progress.

If your Liferay administrator enables the Knowledge Base app's source URL feature (more on this in a moment) and an article has an assigned source URL, an *Edit on GitHub* button (

) appears to the right of the article's title. This button lets users access the article's source location. You can use this feature to encourage users to contribute fixes or improvements to articles. If you're interested in this feature, you can direct your administrator to follow the instructions in System Settings.

Great! Now you know the basics of how Knowledge Base Display works. Next, you'll learn how to let users choose between different article sets in different folders, without giving them access to the app's Configuration menu.

Displaying Different Article Sets As an administrator, say that you've used folders to aggregate similar articles, and you want to provide an easy way for users to switch between these sets of articles. The Knowledge Base Display's content folder feature adds a selector to the top of the navigation that lets users switch between article sets.

To set up content folders follow these steps:

1. Add a folder in Knowledge Base. Then create sub-folders in this folder. These sub-folders are the *content folders*.

2. Add articles to each content folder.

3. Select *Configuration* from Knowledge Base Display's *Options* menu. In the *Setup → General* tab, select the content folders' parent folder and click *Save*.

The content selector's values reflect the names of your content folders. Select one to view its articles. You can also add a common prefix to the names shown in the selector:

1. Navigate to Knowledge Base Display's *Configuration* dialog and select the *Setup → Display Settings* tab.

2. Enter the prefix into the *Content Root Prefix* field and click *Save*.

Awesome! Now you know how to use Knowledge Base Display to display several sets of articles. Next, you'll learn how to use the Knowledge Base Article app to display only individual articles.

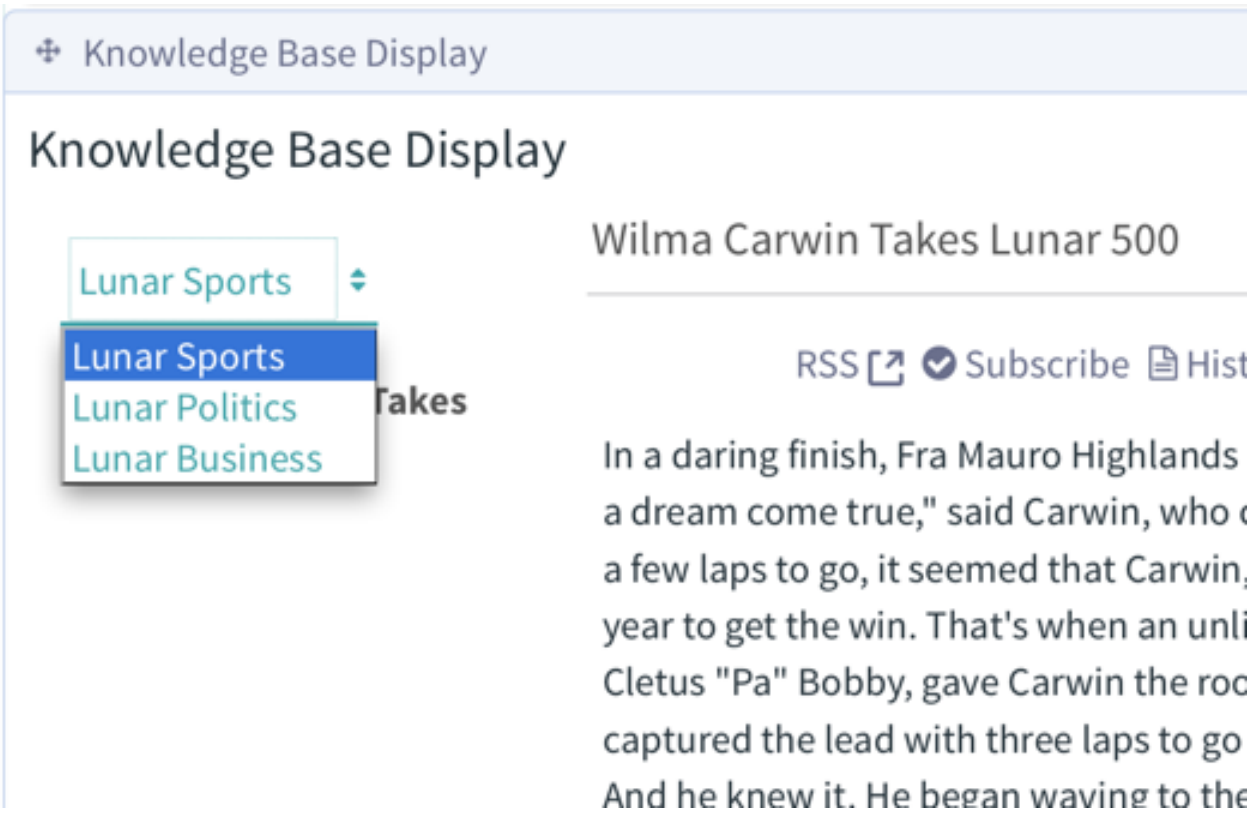


Figure 19.45: Knowledge Base Display's content folder feature lets users switch between different sets of articles.

Knowledge Base Article

The Knowledge Base Article app lets you show a single article's content. It even shows abstracts of child articles. You can add multiple instances of the Knowledge Base Article app to a page, and each one can show a different article.

To display an article follow these steps:

1. Click the *Please configure this portlet to make it visible to all users* link, choose an article, and click *Save*.

2. Refresh the page to display the article.

   You can also use the *Setup* → *General* tab of the app's *Configuration* menu to choose an article to display.

Knowledge Base Article shares the same UI as the Knowledge Base Display to display and manage its articles. Please refer to the Knowledge Base Display section for a detailed description of the app's UI.

Now that you know how to publish individual articles with the Knowledge Base Article app, you can learn how to selectively display articles with the Knowledge Base Section app next.

Knowledge Base Section

Note: as of Knowledge Base 3.0.0, the Knowledge Base Sections app is deprecated and replaced by Liferay Categories.

Figure 19.46: The Knowledge Base Article app is great at displaying individual articles.

The Knowledge Base Section app lets you publish articles associated with a specific topic (section). For example, a news site might have the sections *World*, *Politics*, *Business*, and *Entertainment*.

To use sections, your Liferay administrator must configure the feature in the Knowledge Base app's properties within Liferay DXP's System Settings. Once the feature is enabled, he or she must specify in the System Settings any section names you want to use. This process is covered in more detail in the System Settings section.

Multi-select boxes in the Knowledge Base app's Add Article and Edit Article screens let you define the section(s) the articles belong to. You can add multiple instances of the Knowledge Base Section app to a page and each one can display articles from any number of sections. You can choose to display article titles or abstracts. You can also define whether to show pagination or section titles.

Follow these steps to configure an instance of the Knowledge Base Section app:

1. Select *Configuration* from the Knowledge Base Section's *Options* menu.

2. Select the section or sections (by holding down the shift key) that you want to use and click *Save*.

3. Close the Configuration window to see the updates.

The matching articles are displayed in the app beneath their section heading.

Now that you know how to use the Knowledge Base Section app to organize your articles, you can learn how to use the Knowledge Base Search app to quickly locate your articles next.

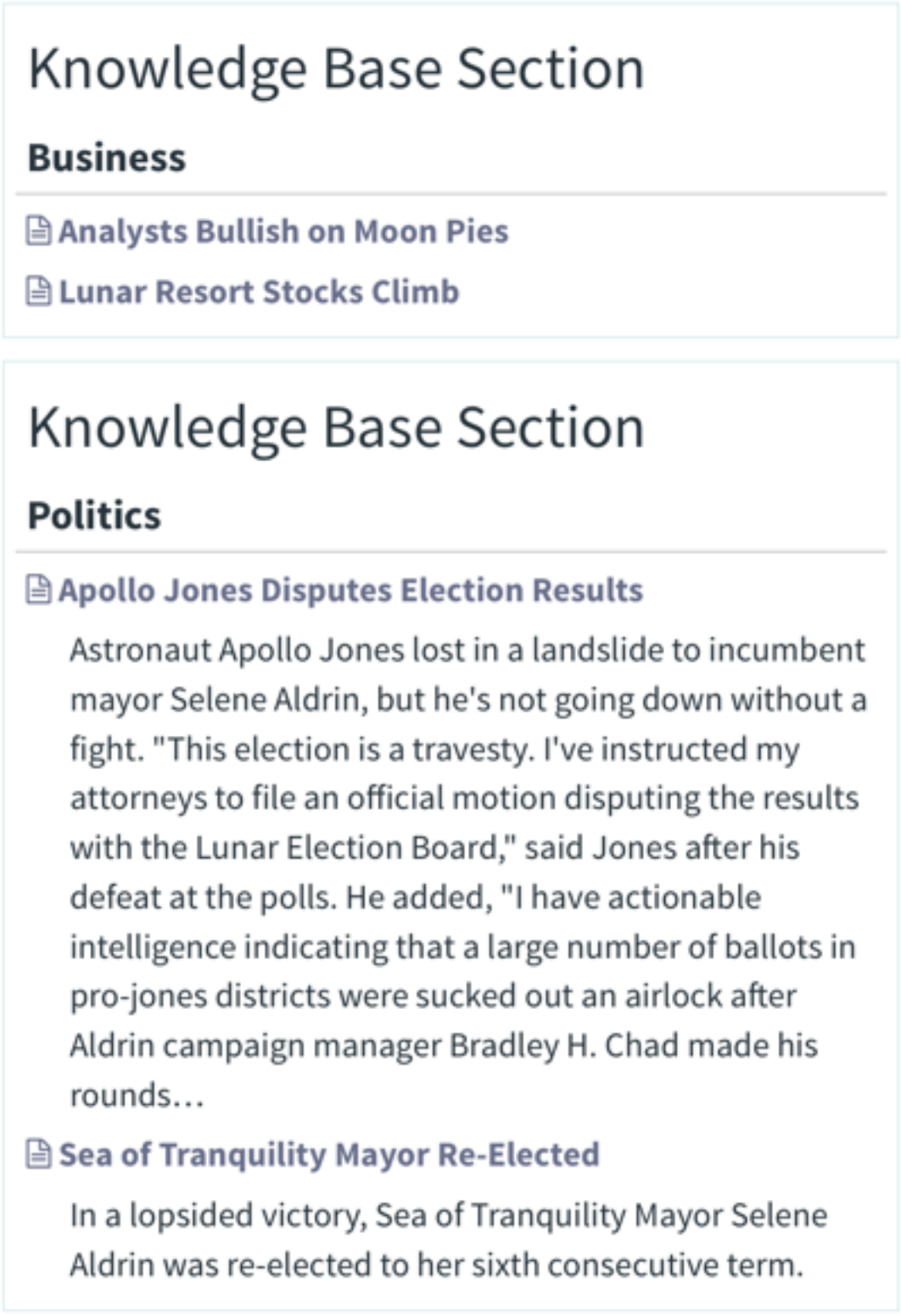


Figure 19.47: Here's an image of Knowledge Base Section apps on a page.

Knowledge Base Search

Note: as of Knowledge Base 3.0.0, the Knowledge Base Search app is deprecated and replaced by Liferay Search.

Even though the Knowledge Base can show the structure of its articles, it may be difficult to find exactly what you're looking for by browsing. That's where the Knowledge Base Search app comes in.

Enter your search term and press the *Search* button. The results are displayed in a table with the following criteria for each matching article:

- title
- author
- create date
- modified date
- number of views

You can select the criteria to display in the app's Configuration dialog.

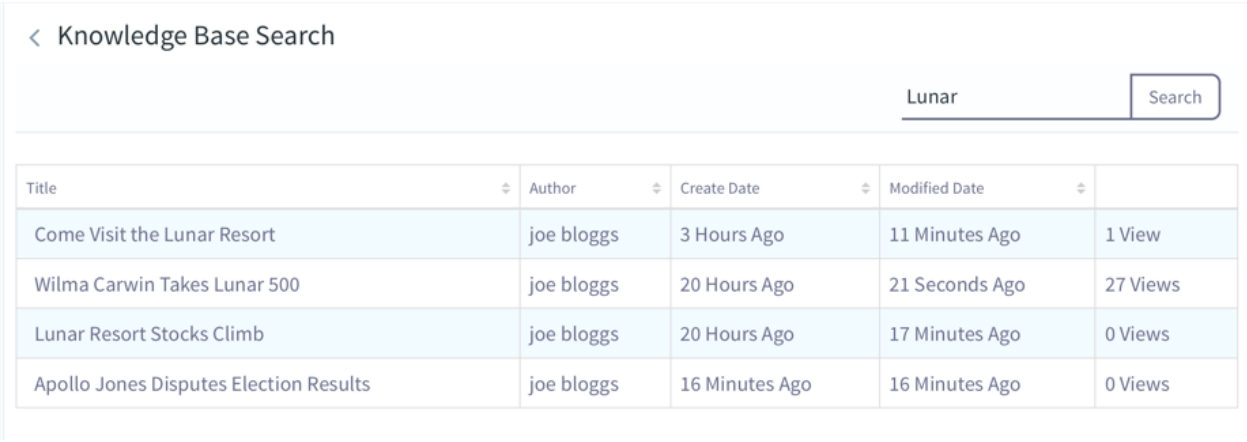


Figure 19.48: : The Knowledge Base Search app helps you search the knowledge base for keywords.

Next, you'll learn how to create new articles and update existing articles by importing them from Markdown source files.

Importing Articles from Markdown Source Files

As mentioned earlier, the Knowledge Base app can import articles in bulk. This lets you have an offline process where articles are prepared ahead of time before they're published. Articles are imported into the Knowledge Base as Markdown files. Markdown is a text-only file format that is easy to read, yet supports all the things you'd need to do to format your articles.

Note: To import articles, your role must be granted the *Knowledge Base → Resource Permissions: Import Articles* permission.

The Knowledge Base supports a Markdown dialect known as Multi-Markdown. This dialect extends the original Markdown with features like table formatting, image captions, and footnotes.

For the Knowledge Base to import your Markdown articles, they must adhere to these requirements:

- All source files must use the `.markdown` or `.md` extensions.
- Articles must start with a top-level header (e.g., `# Some Heading ...`).
- Each header must have an associated, unique ID for the article's friendly URL title and for anchor tags in the article's sub headers. Here's an example of a top-level header that correctly specifies an ID:

  ```
  # Some Heading  [](id=some-heading)
  ```

Below is Markdown source text for a simple example article:

```
# The Moons of Mars [](id=the-moons-of-mars)

As you look up from your chaise lounge, you're sure to see our neighboring
planet Mars. Did you know that Mars has two moons? You might have to break
out a pair of binoculars to see them.

Its two moons are aptly named after the two sons of mythical Roman god Mars.
Their names are Phobos and Deimos.
```

In the first line above, notice the header's ID assignment `id=the-moons-of-mars`. On import, the ID value becomes the Knowledge Base article's URL title.

Markdown is something of a standard: there's Github Flavored Markdown, a proposed common Markdown syntax, forums that support Markdown (reddit, StackExchange, and others), Markdown editors, and an IETF draft for making it an official Internet media type (text/markdown). Why is there so much interest in Markdown?

1. It's readable. Even if you don't know Markdown, you can read it without having to filter out the syntax.
2. It gets out of a writer's way. You don't have to worry about mousing to various icons to change text into a heading or create bulleted lists. Just start typing. The syntax is very intuitive.
3. There are tools to convert it to many other formats, though it was designed to convert to HTML. If your articles are in Markdown, it's straightforward to publish them to the web, mobile formats (Kindle, ePub), and print.
4. Since it's only text, you can use existing tools to collaborate on that text. Using services like GitHub, people can contribute to your articles, and you can see all the changes that have been made to them.

Once your markdown files are ready to import, you must zip them up with their accompanying image files. This is covered next.

Zip File Requirements The importer supports article hierarchies, so Markdown files can be specified anywhere in the ZIP file's directory structure. They can be nested in any number of folders. Image files are the only files supported for attachments.

Note: Imported articles are independent of the workflow settings. This means that **imported articles are automatically approved.**

Only users with the *Import Articles* permission assigned to their role are able to import articles. This permission can be assigned manually through *Control Panel → Users → Roles*. If you've upgraded from 6.2, you can also assign this role to every role that was already able to add articles with a command from the Gogo shell.

Open the Gogo shell: `telnet localhost 11311`. Type `knowledgeBase:addImportArticlePermissions` and hit enter.

The ZIP file's articles are imported in file order (alphanumerically). To designate an article's priority, add a numeric prefix to its file name. For example, the priorities for articles named `01-file.markdown` and `02-file.markdown` become `1.0` and `2.0`.

To designate an article to be the *parent* of all other articles in the same source folder, end its file name with `-intro.markdown`. This creates a parent-child hierarchy. You can use the prefix `00` for parent articles to place them at the top of the folder's file order. The importer uses the numeric prefix of an intro file's folder as its article priority.

Here's the underlying logic for the `00` prefix:

- A file prefix of `00` for a non intro file assigns the resulting article's priority to `1.0`.
- A file prefix of `00` for a top-level intro file sets the article's priority to the first folder numeric prefix found that is `1.0` or greater.

This convention lets you specify priorities for top-level (non-child) articles in your hierarchy.

When importing, keep the checkbox labeled *Apply numerical prefixes of article files as priorities* selected. If a file doesn't have a prefix, its article gets the next available priority (the highest current priority, plus one).

Below is an example ZIP file structure that demonstrates the features mentioned so far:

ZIP File Structure Example:

- `01-winter-events/`
  - `00-winter-excursions-intro.markdown`
  - `01-star-dust-snow-shoeing.markdown`
  - `02-lunar-alpine.markdown`
- `02-summer-events/`
  - `00-summer-excursions-intro.markdown`
  - `01-lunar-rock-scrambling.markdown`
  - `02-extra-terrestrial-mountain-biking.markdown`
  - `03-summer-olympics/`
    * `00-lunar-olympics-intro.markdown`
    * `01-zero-gravity-diving.markdown`
- `images/`
  - `some-image.png`
  - `another-image.jpeg`

The above ZIP file specifies `00-winter-excursions-intro.markdown` as the parent of its neighboring Markdown files: `01-star-dust-snow-shoeing.markdown` and `02-lunar-alpine.markdown`. Likewise, `00-lunar-olympics-intro.markdown` is the parent of `01-zero-gravity-diving.markdown`. `00-lunar-olympics-intro.markdown` is also the peer of `01-lunar-rock-scrambling.markdown` and `02-extra-terrestrial-mountain-biking.markdown`, and the child of `00-summer-excursions-intro.markdown`.

ZIP Example's Resulting Relationships and Priorities

- `01-winter-events/00-winter-excursions-intro.markdown`

  - **Article:** Winter Excursions
  - **Relationship:** Peer of *Summer Excursions*
  - **Priority: 1.0**

- `01-winter-events/01-star-dust-snow-shoeing.markdown`

  - **Article:** Star Dust Snow Shoeing
  - **Relationship:** Child of *Winter Excursions*
  - **Priority:** 1.0

- `01-winter-events/02-lunar-alpine.markdown`

  - **Article:** Lunar Alpine
  - **Relationship:** Child of *Winter Excursions*
  - **Priority:** 2.0

- `02-summer-events/00-summer-excursions-intro.markdown`

  - **Article:** Summer Excursions
  - **Relationship:** Peer of *Winter Excursions*
  - **Priority: 2.0**

- `02-summer-events/01-lunar-rock-scrambling.markdown`

  - **Article:** Lunar Rock Scrambling
  - **Relationship:** Child of *Summer Excursions*
  - **Priority:** 1.0

- `02-summer-events/02-extra-terrestrial-mountain-biking.markdown`

  - **Article:** Extra Terrestrial Mountain Biking
  - **Relationship:** Child of *Summer Excursions*
  - **Priority:** 2.0

- `02-summer-events/03-summer-olympics/00-lunar-olympics-intro.markdown`

  - **Article:** Lunar Olympics
  - **Relationship:** Child of *Summer Excursions*
  - **Priority:** 3.0

- `02-summer-events/03-summer-olympics/01-zero-gravity-diving.markdown`

  - **Article:** Zero Gravity Diving
  - **Relationship:** Grandchild of *Summer Excursions*
  - **Relationship:** Child of *Opening Ceremonies*
  - **Priority:** 1.0

ZIP files must meet the following requirements:

- Each Zip file must end in the suffix `.zip`.
- Each Zip file must contain at least one Markdown source file, optionally organized in folders.
- All referenced image files must be in a folder named `images` in the Zip file's root.
- Image files must be in a supported format and must use the appropriate file extensions. Supported extensions are `.bmp`, `.gif`, `.jpeg`, `.jpg`, and `.png`. They're specified via an app system setting. For details, see System Settings.

Once you have your article ZIP file, it's time to import it.
Follow these steps to import your ZIP file:

1. Navigate to *Site Administration* → *Content* → *Knowledge Base* → *Articles*.
2. Click on *Add* → *Import* to bring up the importer page.
3. Browse to the location of your file, and in most cases leave the checkbox for the article priorities checked, and then click *Save*.

Your file is uploaded, and the importer converts each source file's Markdown text to HTML, applying the HTML to the resulting article. Any image files that are referenced in an article and included in the ZIP file are imported as attachments to the article.

Figure 19.49: : Selecting *Add* → *Import* in Knowledge Base brings up the interface for selecting a ZIP file of Markdown source files and images to produce and update articles in your Knowledge Base.

In addition to source files and images, you can configure a base source URL system setting for the importer that specifies your source file's repository location. Each article's *Edit on GitHub* button (if enabled) takes the user to the source location. The importer prefixes each file's path with the base source URL. This constructs a URL to the article's repository source location; it looks like `[base URL]/[article file path]`. Here's an example base source URL:

```
https://github.com/liferay/liferay-docs/blob/master/develop/tutorials
```

The source URL constructed from this base URL and article source file `folder-1/some-article.markdown` would be:

```
https://github.com/liferay/liferay-docs/blob/master/develop/tutorials/folder-1/some-article.markdown
```

You specify the base source URL in a file called `.METADATA` in the Zip file's root folder. The importer treats the `.METADATA` file as a standard Java properties file and uses the base source URL to construct the source URL for all of the Zip file's resulting articles.

To use the source URL feature, your Liferay administrator must enable it via the Knowledge Base app's System Settings.

Now that you know how to import articles into the Knowledge Base, you may have some questions about the importer. Some of the common questions and answers are covered in the next section.

Importer FAQs

- **What happens when I import an existing article?** The importer checks if the source file's leading header ID (e.g., `# Some Heading [](id=some-heading)`) matches the URL title of any existing article in the Knowledge Base folder. If a match is found, the importer replaces the article's content with the incoming content converted from the source file. If no match is found, a new article is created.
- **Do I need to import all of a Knowledge Base folder's articles, even if I only want to create a new article or update a subset of the folder's current articles?** No. You can import as many or as few new or modified articles as you like.
- **Does the importer remove articles?** No. The importer only creates and updates articles. It doesn't delete any existing articles. To delete an article, you must manually do so via the Knowledge Base apps.
- **Can I update an article's priority?** Yes. You can use the file/folder prefix convention and re-import the article to update its priority.
- **If I change an article's title, should I also change its header ID?** It depends on whether you've already published your article. If it hasn't been published, then there are no public links to it, so it's fine to change the header ID. If the article is already published, you must decide whether it's worth breaking existing links to the article, and worth having search engines rediscover and re-rank your article based on its new friendly URL. The new friendly URL is based on the new header ID.

Congratulations on mastering the art of creating and importing Markdown files to produce Knowledge Base articles!

Next, Liferay administrators can learn how to override the Knowledge Base app's System Settings to enable and configure features.

System Settings

Liferay administrators can configure the Knowledge Base across sites using System Settings. To access its settings, open the Control Panel, navigate to *Configuration → System Settings*, and click on the *Collaboration* link. There are five sections of Knowledge Base configuration settings:

- *Knowledge Base Article*
- *Knowledge Base Display*
- *Knowledge Base Search*
- *Knowledge Base Section*
- *Knowledge Base Service*

The *Knowledge Base Service* section's settings apply defaults to all the Knowledge Base apps and to the Knowledge Base in Site Administration. The other sections apply to specific Knowledge Base apps and override the *Knowledge Base Service* defaults.

Important: Advanced configuration of the Knowledge Base application's system settings should only be performed by a Liferay administrator.

The Knowledge Base has several optional features that are disabled by default, but can be enabled and configured from System Settings. These include source URL, import file conventions, new article priority increment, and sections. Now you'll learn how to configure these features by editing the Knowledge Base System Settings.

Source URL Settings The source URL settings define the source location of importable Markdown files. This is intended to point to a source repository where the files are stored. GitHub is assumed as the default. Once defined, the Knowledge Base displays a button (default label is *Edit on GitHub*) above each displayed article. Users can click the button to navigate to an article's source location.

The source URL settings are accessible in the *Knowledge Base Service* System Settings section. To enable using the source URL, check the *Source URLEnabled* checkbox.

To change the source URL button's label, specify a new value for the *Source URLEditmessage key* setting. Best practice is to specify the value as a language key. For example, if you create a language key `edit-on-bitbucket=Edit on Bitbucket`, you can specify that language key as the button's new label:

```
edit-on-bitbucket
```

Alternatively, you can specify the label explicitly:

```
Edit on Bitbucket
```

Next, you'll learn how to modify the file conventions for the importer.

Importer File Convention Settings These settings define the supported file extensions, the suffix for parent files, and the image folder's path within the import Zip files. These settings are accessible in the *Knowledge Base Service* System Settings section.

The following settings specify the supported file extensions for the importer:

- *Markdown importer article extensions* sets the supported article extensions. The default values are `.markdown` and `.md`.
- *Markdown importer image file extensions* sets the supported image file extensions. The default values are `.bmp`, `.gif`, `.jpeg`, `.jpg`, and `.png`.

To modify the supported file extensions follow these steps:

1. Click the + or - button next to the setting to add or remove a supported file extension respectively.
2. Enter a new value, if adding an extension.
3. Click *Save*.

These settings define additional article configuration options for the importer:

- *Markdown importer article intro* sets the article parent file suffix. The default value is `intro.markdown`.
- *Markdown importer image folder* sets the image folder path the importer looks for in the article Zip file. The default path is `/images`.

- *New Article Priority Increment Setting* sets the floating point value by which priorities of new articles are incremented. By default they're incremented by `1.0`. To disable this increment so that articles get a flat value of `1.0`, deselect the checkbox for the setting *Article increment priority enabled*.

  Alternatively, you can enable or disable the article increment priority feature for each app in the corresponding Knowledge Base app configuration menu in System Settings.

If you're using the Knowledge Base Section app, you'll want to set some section names. Let's explore that next.

Section Names Setting The section names setting lets you specify names of arbitrary topics to attribute to articles. Using the Knowledge Base Section app, you can display one or more *sections* (groups) of articles. To use sections, you must first define them in the System Settings for the *Knowledge Base Section* app.

Follow these steps to make new sections available:

1. Navigate to the *Knowledge Base Section* configuration menu from *System Settings* → *Collaboration*.
2. Click the plus button next to the *Admin KBArticle sections* setting to add a new field for each section you want.
3. Enter a name for each new section and click *Save*.

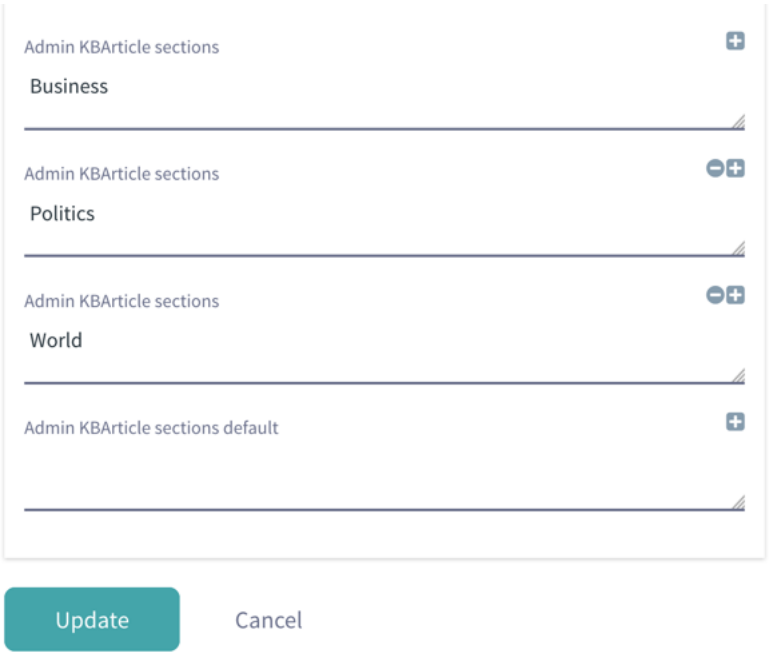


Figure 19.50: : Adding a new field for the *Admin KBArticle sections* setting creates a new section for the Knowledge Base Section app. These fields create sections *Business*, *Politics*, and *World*.

Once your sections are added, you can follow the steps in the Knowledge Base Section app section to learn how to use them.

This concludes the short tour of a sampling of Knowledge Base configuration settings.

For a full list of the available System Settings for Knowledge Base see Knowledge Base Configuration.

You've learned the benefits of publishing articles using the Knowledge Base app. You've seen how easy it is to create, organize, and manage articles, and you've learned various ways to present articles in the Knowledge Base Display, Knowledge Base Article, and Knowledge Base Section apps. Knowledge Base system settings are also at your service. You can now consider yourself truly *knowledgeable* of Liferay's Knowledge Base app.

19.8 Bookmarking Sites

Many of us enjoy collecting things we value: stamps, comic books, sea shells, shoes, or even vintage musical instruments. You've probably never thought about collecting URLs, even though most of us collect them. Anyone that uses a browser likely has an extensive URL collection in the form of bookmarks. With Liferay's Bookmarks app, your Liferay instance can provide its users with bookmarks that benefit the entire community.

The Bookmarks app lets users and administrators add, edit, delete, export, and import URLs in your Liferay instance. Through the Bookmarks app, users can access web sites and administrators can publish links tailored to specific user segments. Both internal pages as well as external sites can be bookmarked. Next, you'll learn how to create bookmarks and folders in the Bookmarks app.

Creating Bookmarks and Folders

To use the Bookmarks app, you must first add it to a page. Select *Add* → *Applications* → Community, and then drag *Bookmarks* onto the page. By default, the app looks like this:

The Bookmarks app is empty at first. The app greets you with a smiley face and a message that says there are no bookmarks in the app's current bookmarks folder, the *Home* folder. The current folder's name is displayed to the top-left of the area with the smiley face and message. The app displays the *management bar* immediately above the current folder's name. The management bar lets you choose which bookmarks and folders to display, how to display them, and more. The search bar, located to the top-right of the management bar, lets you search for folders and bookmarks in the app.

To see all this functionality in action, you must first add some bookmarks and folders. When you mouse-over the app, another bar appears across the top that shows the app's name, an Add button (), and an Options button ( ). When you click the Add button, a small menu pops up that lets you select *Folder* or *Bookmark*. Selecting *Bookmark* brings up the *Add Bookmark* form. This form lets you add a bookmark by filling in the *Name*, *URL*, and *Description* fields. Below these fields, there are additional options for Categorization, Related Assets, and Permissions, just like in other Liferay apps. See the Publishing Content Dynamically section for more information on this. When you're finished filling in the Add Bookmark form, click *Save* to create the bookmark. If you select *Folder* from the add button, this brings up the *Add Folder* form. This form is very similar to the Add Bookmark form. Fill in its *Name* and *Description* fields, assign any permissions you want to use, and click *Save* to create the folder. When you create folders and bookmarks, they're added to the current folder. Folders can have any number of subfolders.

To import or export bookmarks, mouse-over the app, click the *Options* button next to the Add button at the top of the app, and select *Export/Import*. In the window that appears, you can start a new export or import process. For details, see the article on importing and exporting pages and content.

Next, you'll learn how to manage bookmarks and folders.

Managing Bookmarks and Folders

The Bookmarks app provides several ways to manage your bookmarks and folders. One is the management bar. Checking the checkbox on the left-hand side of the management bar selects all the folders and bookmarks currently displayed in the app. When you do this, the management bar changes to show the number of

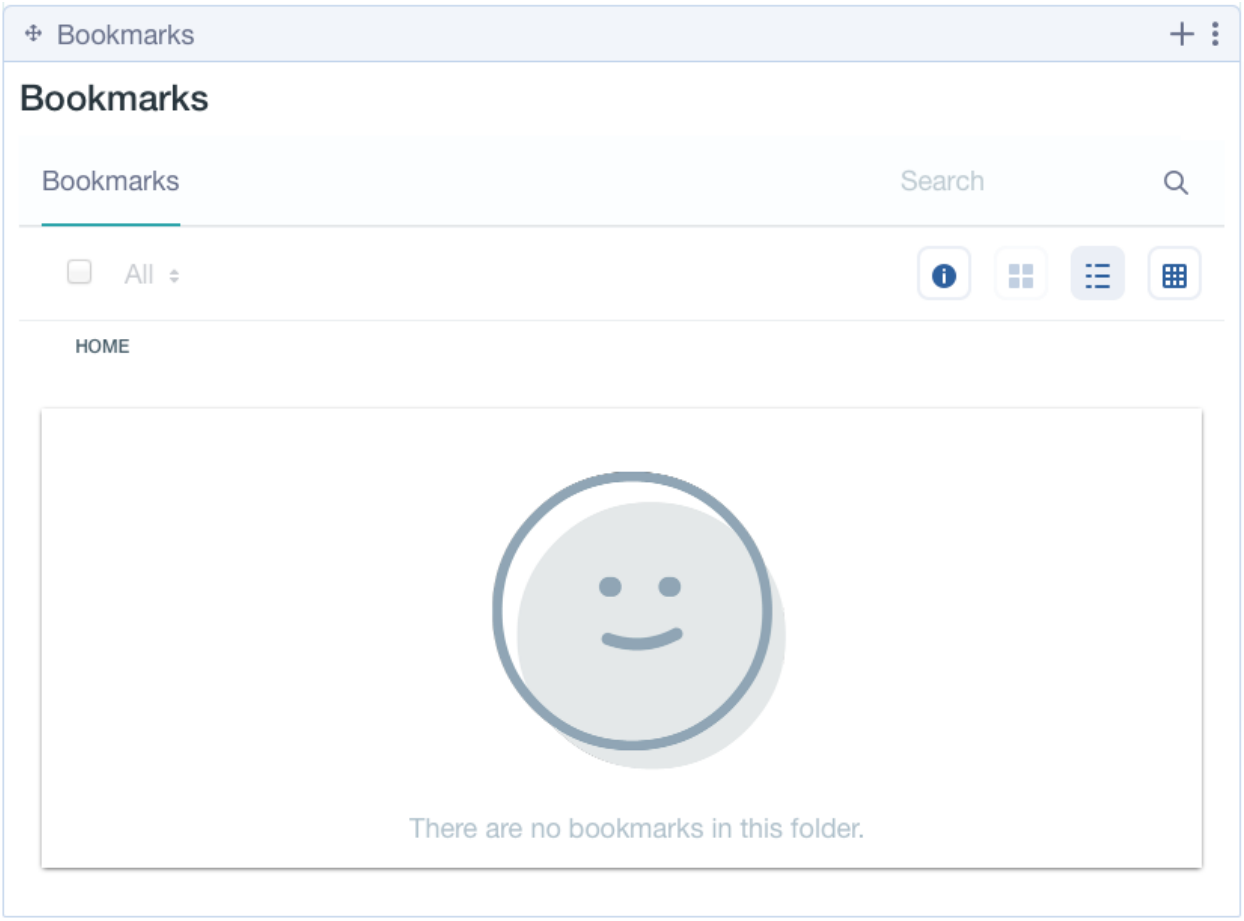


Figure 19.51: Initially, no bookmarks appear in the Bookmarks app.

selected items, and replaces the buttons on its right with an Info button ( ) and a Recycle Bin button ( ). Pressing the Recycle Bin button moves the selected items to the Recycle Bin. Uncheck the checkbox to return the management bar to its initial state. In this state, the selector menu to the right of the checkbox lets you filter the items to display:

- **All:** Display all folders and bookmarks.
- **Recent:** Display only recent bookmarks.
- **Mine:** Display only the bookmarks you created.

The *Info*, *Descriptive* ( ), and *List* ( ) buttons are on the management bar's right side. Clicking the Info button slides in an info panel that displays more information about the current folder and gives you additional management options. To receive notifications when a user adds a bookmark to the folder, subscribe to the folder by clicking the *Star* icon ( ). Click the *Actions* menu ( ) to reveal options for editing, moving, assigning permissions to, or deleting the folder. Note that you can't edit, move, or delete the Home folder.

The management bar's Descriptive and List buttons determine how the app displays folders and bookmarks. Click the Descriptive button to show each item's name, description, number of subfolders and

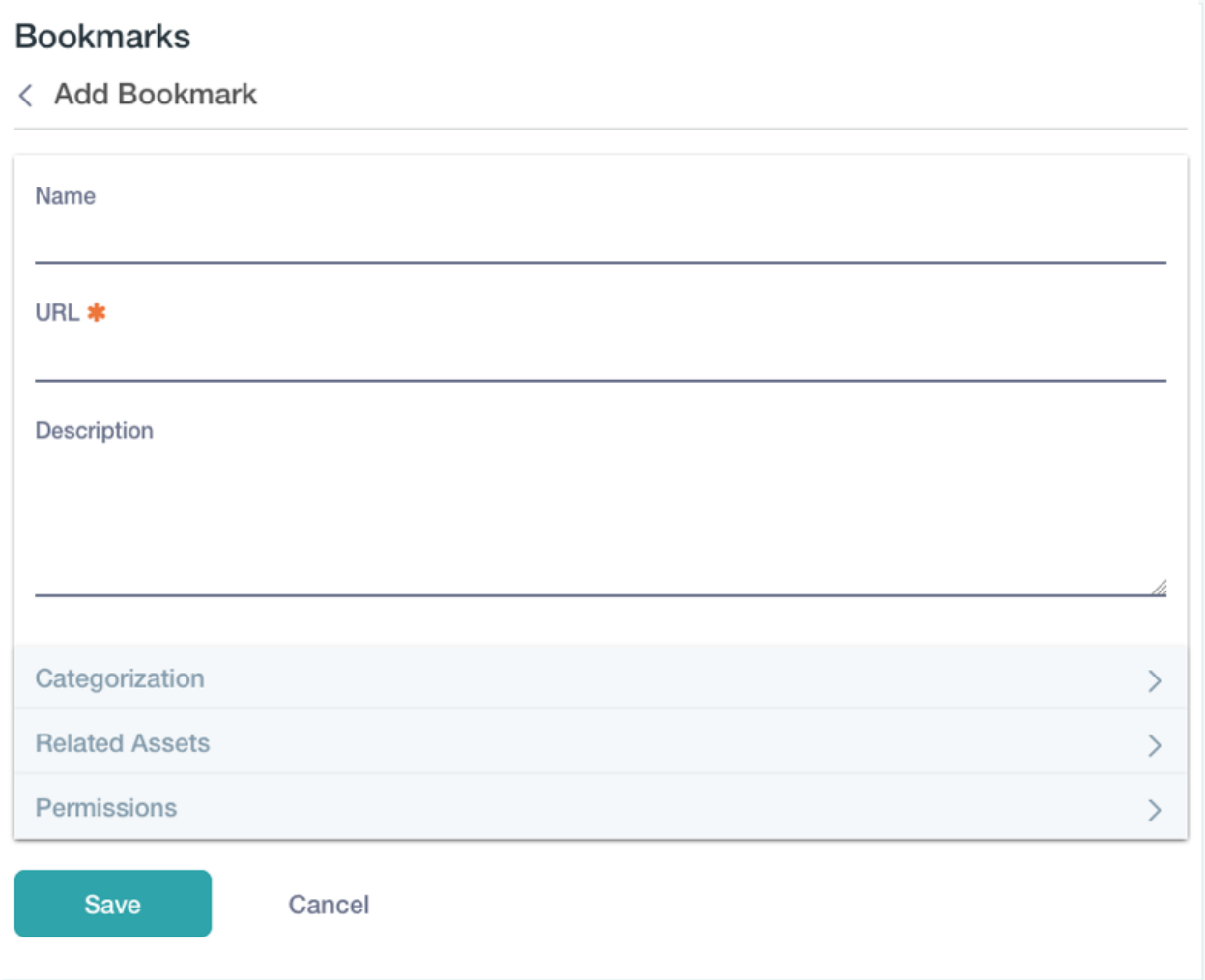


Figure 19.52: The Add Bookmark form lets you create a bookmark.

bookmarks (folders), and number of visits (bookmarks). The app displays items this way by default. To instead show items in a simple list, click the List button.

You can also manage each item from its *Actions* menu. You can use this menu to edit, move, assign permissions to, or delete each item. If the item is a bookmark, you can also subscribe to it in its actions menu. Subscribing to a bookmark means that you receive a notification when the bookmark changes.

When moving an item, the Bookmarks app presents you with the Move Entries form. In this form, the item's new folder is selected for you automatically. Click *Move* to move the item to this folder. To choose a different folder, click *Select*. The window that appears lets you navigate your folder hierarchy, beginning with the Home folder. Click any folder's name to view its subfolders. The text at the top of the window indicates your position in the hierarchy. You can click the folder names in this text to move back up the hierarchy. To select a subfolder, click its *Choose* button. To choose the folder you're currently browsing, click the *Choose This Folder* button. To create a new folder, click the *Add Subfolder* button. Note that when you're in the Home folder, this is instead the *Add Folder* button. The window closes once you choose a folder, returning you to the Move Entries form. Click *Move* to move the item to the selected folder.

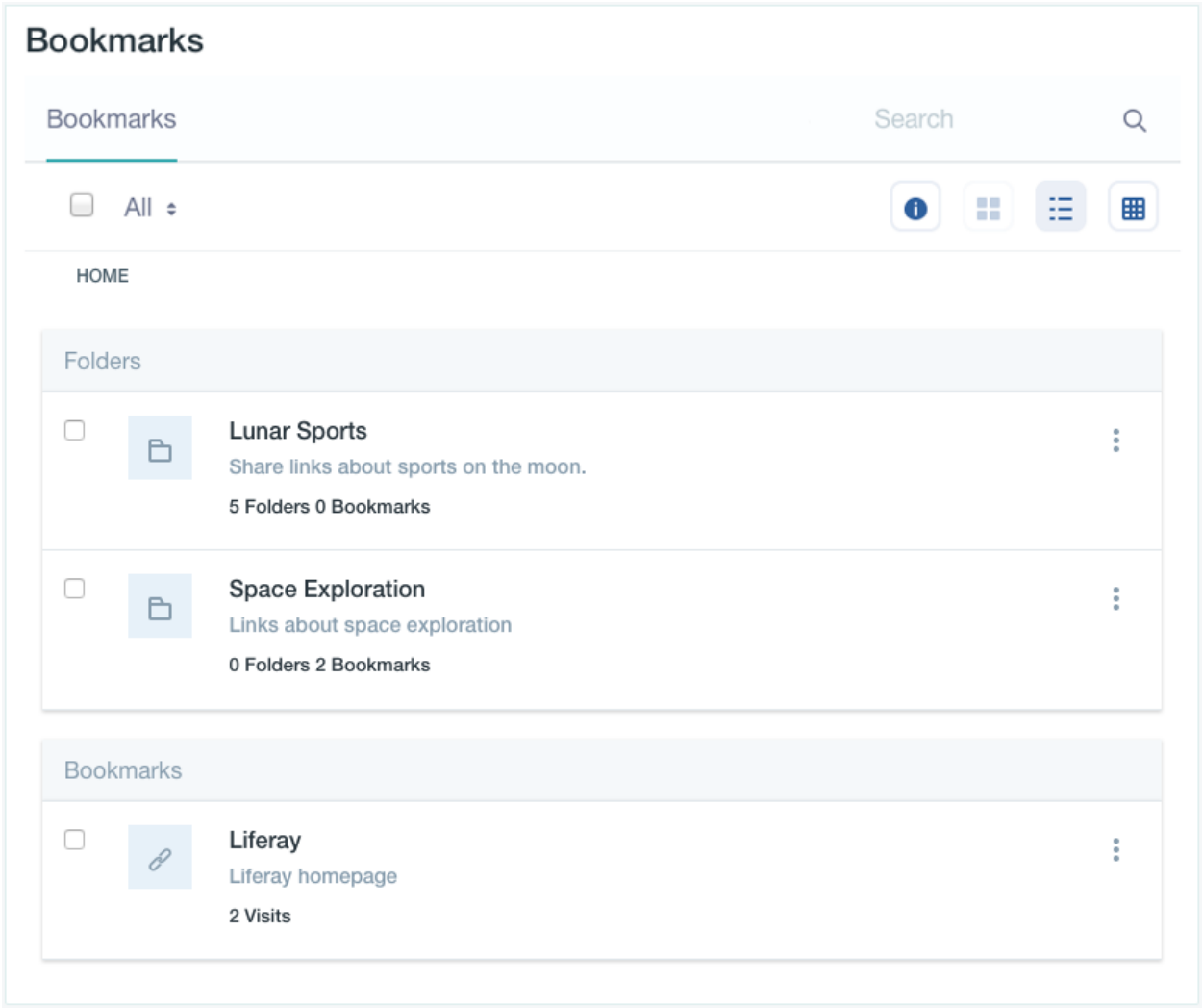


Figure 19.53: The Bookmarks app displays your folders and bookmarks.

19.9 Summary

Liferay's Bookmarks app gives users and administrators a straightforward way to add and manage bookmarks in your Liferay instance. This lets everyone share and access URLs relevant to the community, including URLs for internal pages. Since bookmarks can be organized in folders nested to unlimited levels, the Bookmarks app is suitable for Liferay instances with any number of users.

19.10 Sharing Status with Microblogs

Liferay's Microblogs Status Update and Microblogs apps let you post, receive, and share short, 150 character status updates from your personal pages in Liferay. This gives you a quick and simple way to interact with your social connections and other users. Microblogs Status Update is a very simple app that lets you post a status update and displays your most recent post. The Microblogs app, in addition to this functionality, also lets you:

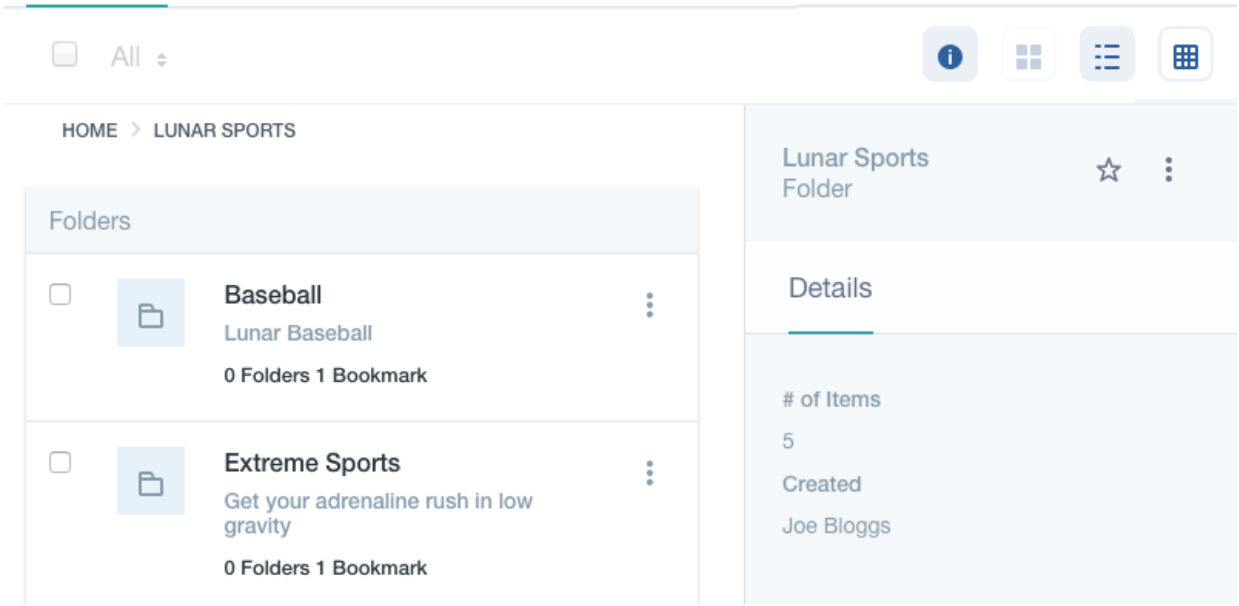


Figure 19.54: A folder's info panel shows more information about the folder, and lets you take additional actions.

Figure 19.55: Choose the folder to move the item to.

- View all your posts and those of your social connections.
- Comment on and share posts.
- View posts that mention you or contain a specific hashtag.

Before delving into these details, you should first understand where in your Liferay instance you'll use these apps.

Understanding Where to Use the Apps

Because Microblogs Status Update and Microblogs broadcast your personal status, you must use them from your Liferay personal site's pages. By default, these pages are called *My Profile* and *My Dashboard*. You can access both from the user menu. My Profile is public and serves as your public profile. Users and guests can view any apps you add to this page, although app appearance and functionality are typically limited. My Dashboard is private and by default contains the *My Sites* app. You can add other apps to this page to help you, and only you, manage your Liferay experience.

You can add the Microblogs Status Update and Microblogs apps to both pages, but the apps behave differently depending on where they are and who sees them. If you put them on a public page like My Profile, only logged in users can see your posts, and posting is disabled. After all, you wouldn't want others to make posts on your behalf! The internet is crazy enough without granting others that kind of power. Logged in users can, however, comment on posts in the Microblogs app. Guests can see the apps, but not the posts. Things get a great deal more interesting when you add the apps to a private page that only you can access, like My Dashboard. In this case, you unleash the apps' full power. Next, you'll learn how to wield this power.

Using Microblogs Status Update and Microblogs

As mentioned, Microblogs Status Update is a very simple app that lets you post your status, and displays only your most recent post. That's it. You can even configure it not to display this post. To do so, mouse over the app, click the *Options* button (⋮) that appears on top of the app, and click *Configuration*. In the *Setup* tab, uncheck *Display Most Recent Status* and click *Save*. To post your status, enter it into the *Update Your Status* field and click the *Post* button that appears. You can also select who can view your post from the *Viewable By* menu next to the Post button. You can select *Everyone*, *Connections*, or *Followers*. For more information on connections and followers, see the section Creating a Social Network.

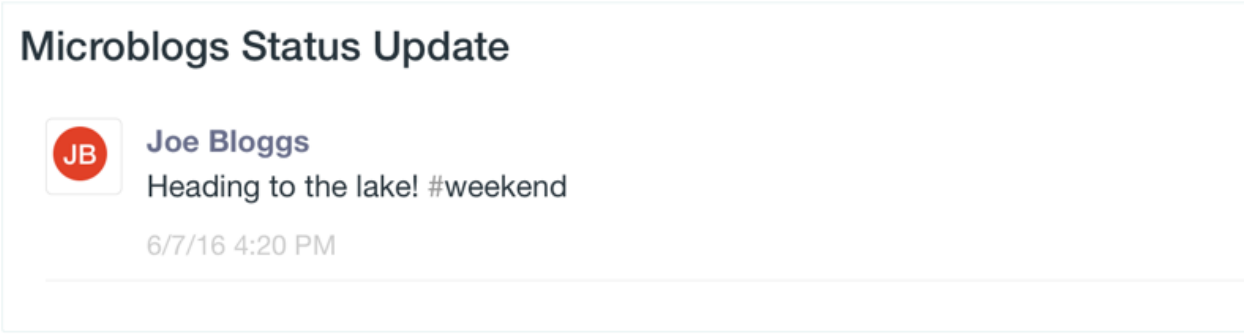


Figure 19.56: The Microblogs Status Update app lets you post your status, and displays your most recent post.

There are two special symbols that you can use in your posts. The hashtag symbol (#) turns the word that follows it into a hashtag. When viewing a post, hashtags appear as links. You can click on any hashtag to

view a list of posts that contain the same hashtag. Note that hashtags are distinct from Liferay tags–only Liferay tags are indexed and made available for search. The other special symbol you can use is the mentions symbol (@). When writing a post, typing the @ symbol brings up a list of connections to select. Once you select a connection and publish your post, the connection's username becomes a link to their public profile page (their My Profile page).

The Microblogs app is effectively the Microblogs Status Update app on steroids. You can post your status the exact same way. The Microblogs app, however, displays *all* your posts in the *Timeline* tab. The posts of your connections and those you follow also appear in the Timeline. You can also comment on any post, edit your posts, and repost others' posts. The *Mentions* tab displays any posts that mention you. If you click a hashtag in any post, a new tab opens in the Microblogs app that shows all the posts containing that hashtag.

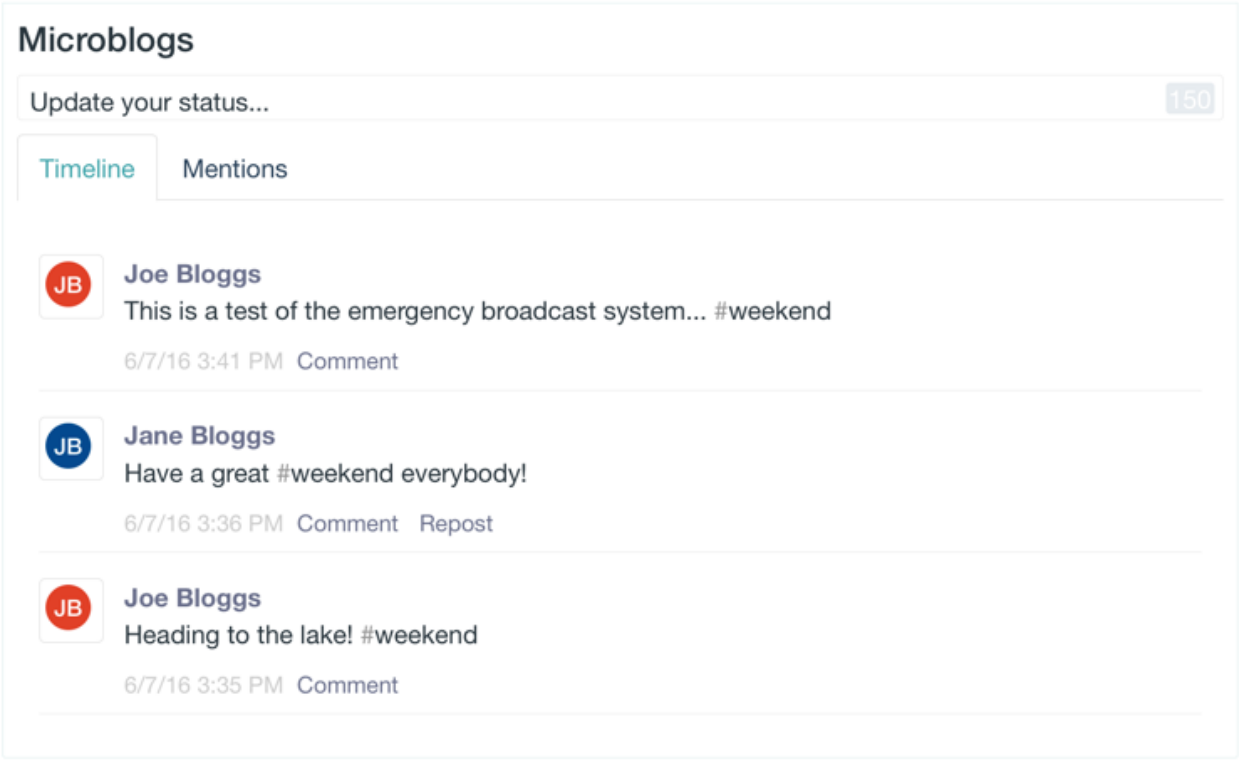


Figure 19.57: The Microblogs app displays all your status posts, along with those of your social connections. It also lets you view any posts you're mentioned in.

As you've seen, the Microblogs Status Update and Microblogs apps are simple apps that keep you in touch with your fellow Liferay users. Next, you'll learn how to fully leverage Liferay's social apps to build a social network in your Liferay instance.

19.11 Inviting Members to Your Site

You've built a new site and now you need to get the word out about it. What better way to get the word out about your site then to invite users to join it. The Invite Members app lets you to do just that!

The Invite Members app is only visible to site administrators. It allows administrators to send invitations to users to join the site. Clicking the *Invite members to this site* button brings up the interface for sending out invitations.

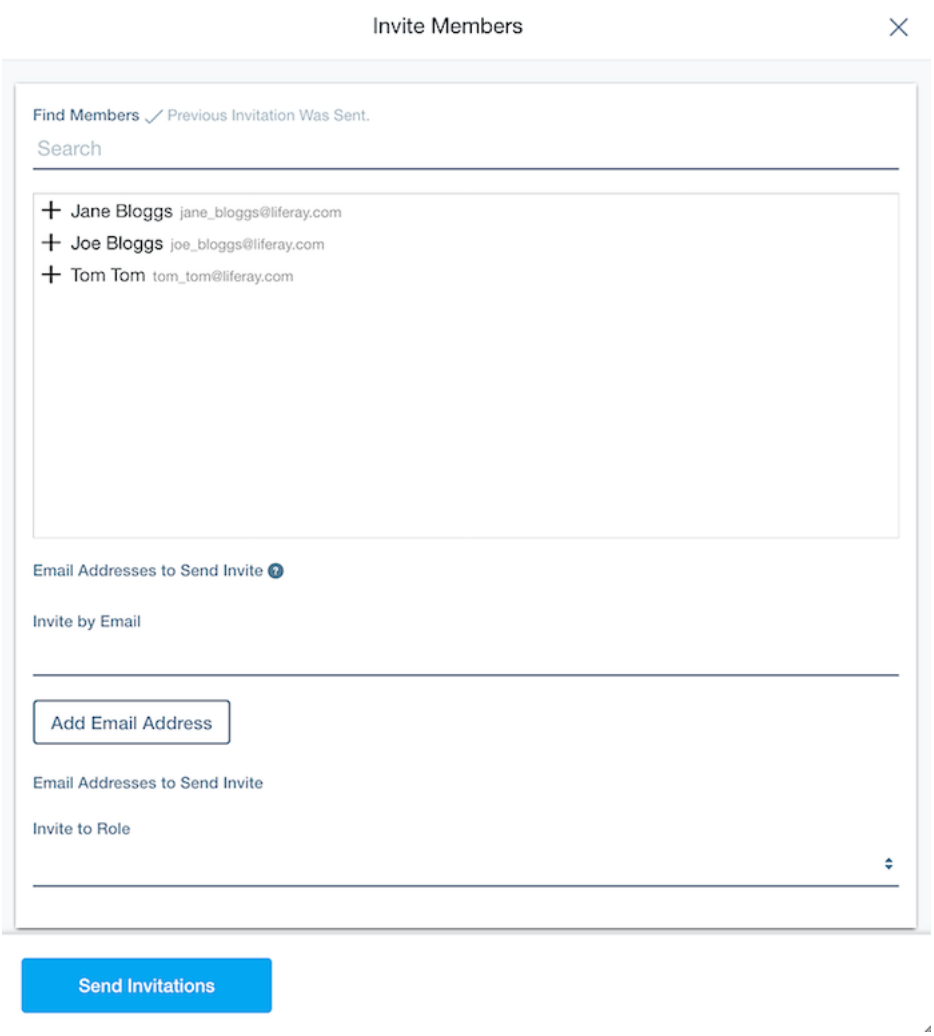


Figure 19.58: You can invite users by clicking the add sign next to the user's name.

Click the plus sign next to a user or click the *Add Email Address* button to add a user to the invite list. Users that have already been invited but have not yet responded are listed with a check mark next to their names. You can also invite users to the *Site Owner*, *Site Content Reviewer*, and *Site Administrator* roles for your site by selecting that role under the *Invite to Role* heading. Once you've added all the users you want to invite and have selected any roles they should have, click the *Send Invitations* button to invite them.

The site invitation shows up under the *Requests List* tab on the user's *Notifications* page. The user can then choose to *Confirm* or *Ignore* the invitation.

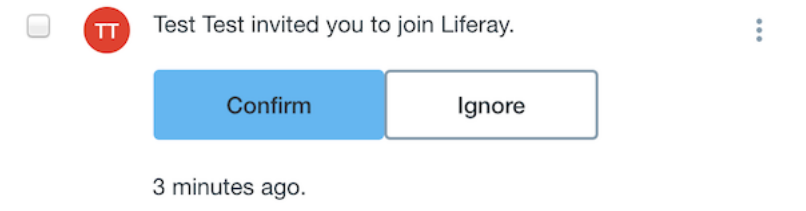


Figure 19.59: You can confirm or ignore the invitation.

Note: The Invitation app is also available as a deprecated app on the Liferay Marketplace. This app, unlike the Invite Members app, lets you send an email invitation to a list of specified email addresses. You can customize the email message for the invitation through the portlet's Configuration menu.

This app is deprecated as of version 1.0.0.

When the invitation is confirmed, the user is made a member of the site and assigned the roles you defined.

19.12 Sending Messages with Private Messaging

The Private Messaging app lets you send email-like messages to other users. Its interface is similar to an email client. Messages are aggregated by thread, with the most recent message displayed. You can mark messages as read or unread or delete them.

Note: There is a known issue, LPS-69331, which causes the Private Messaging app to be treated as a brand-new module during the initial upgrade process. This issue is fixed in Liferay fixpack de-10.

The Private Messaging app must be installed from Marketplace. Once installed, the app must be added to a user's private *Dashboard* page. Now that you know where you can get the app and where to place it, you're probably eager to write messages. This is covered next.

Composing Messages

Once the app is added to a user's Dashboard page, click the *New Message* button to compose a new message. Just like email, you can enter a subject, a message body, and add attachments. In the *To* field, an autocomplete recipient list of users appears that lets you select who to send your message to. You can select any number of users.

By default, all users are available in this list. You can filter the available list of users by configuring the app. This is covered next.

Configuring System Settings

You can filter the set of available users in the *To* field's autocomplete recipient list by social connection, status, or site membership. This configuration, however, must be done by an administrator who has access to the Private Messaging portlet's system settings.

To access the Private Messaging portlet's system settings, open the Control Panel, navigate to *Configuration* → *System Settings*, and click *Private Messaging Service* under the *Other* tab.

The following settings filter the autocomplete recipient list of users for the Private Messaging portlet:

Autocomplete recipient type: Filters the scope of users that appear in the autocomplete recipient list of the message's *To* field. Note that although this setting restricts the autocomplete recipient list of users, you can still send a private message to a user if you know his or her user name. The default value is `all`. Possible values are listed below:

- `all`: displays all users across your installation in the autocomplete recipient list.
- `connection`: displays only social connections in the autocomplete recipient list.
- `site`: displays only fellow site members in the autocomplete recipient list.

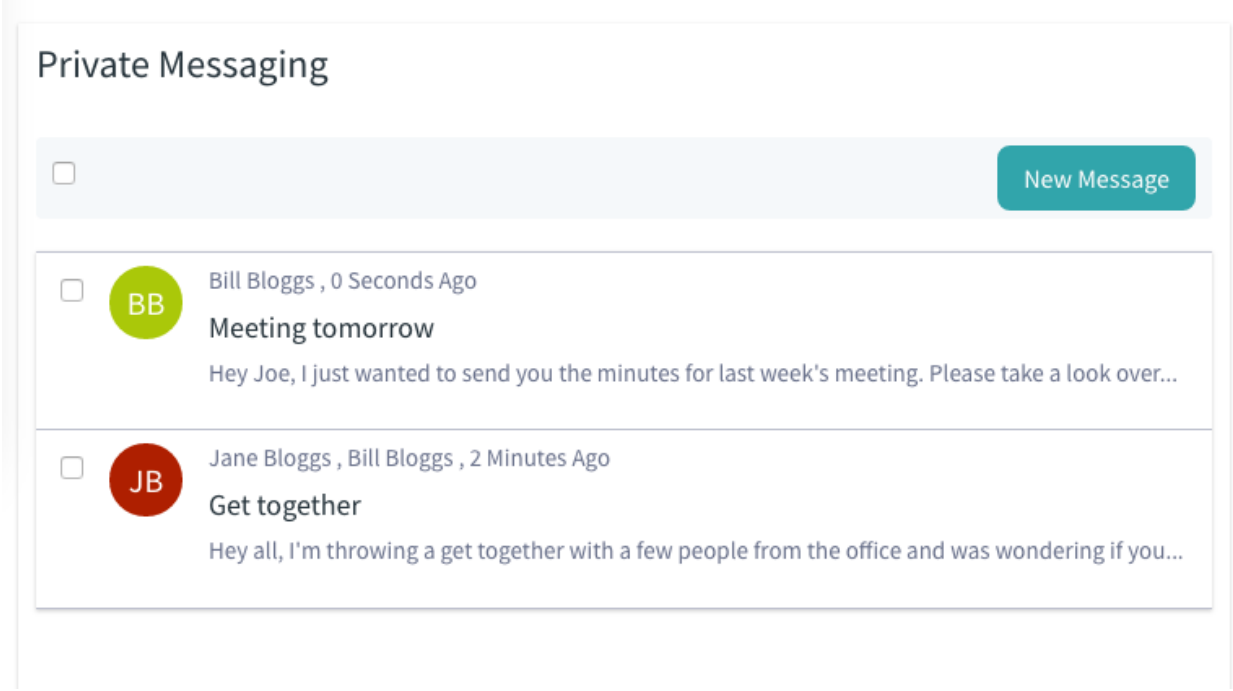


Figure 19.60: The Private Messaging Interface functions just like an email client for use within a Portal instance.

Autocomplete recipient site excludes: Sets specific sites to exclude from the autocomplete recipient list when **Autocomplete recipient type** is set to `site`. This is especially useful if you have a default site that all users belong to, but you still want to filter the autocomplete recipient user list by site. That site is then ignored when determining the users that are available in the autocomplete recipient list.

19.13 Managing Subscriptions

Message Boards, Blogs, and Wikis, oh my! There are a lot of assets that you can subscribe to in Liferay DXP, and keeping track of all your subscriptions can be a job. Thanks to the My Subscriptions app, available on Liferay Marketplace, you can manage all your subscriptions from one app.

Add the app to a page and your subscriptions are listed underneath their corresponding asset type.

Each subscription has an Actions menu that lets you *View*, *View in Popup*, and *unsubscribe* from the asset. You can unsubscribe from multiple subscriptions by selecting the checkbox next to the assets you want to unsubscribe from and clicking the *Unsubscribe* button. Alternatively you can unsubscribe from all subscriptions of a specific asset type by selecting the checkbox for the asset type (e.g. *Message Boards Category*) and clicking the *Unsubscribe* button.

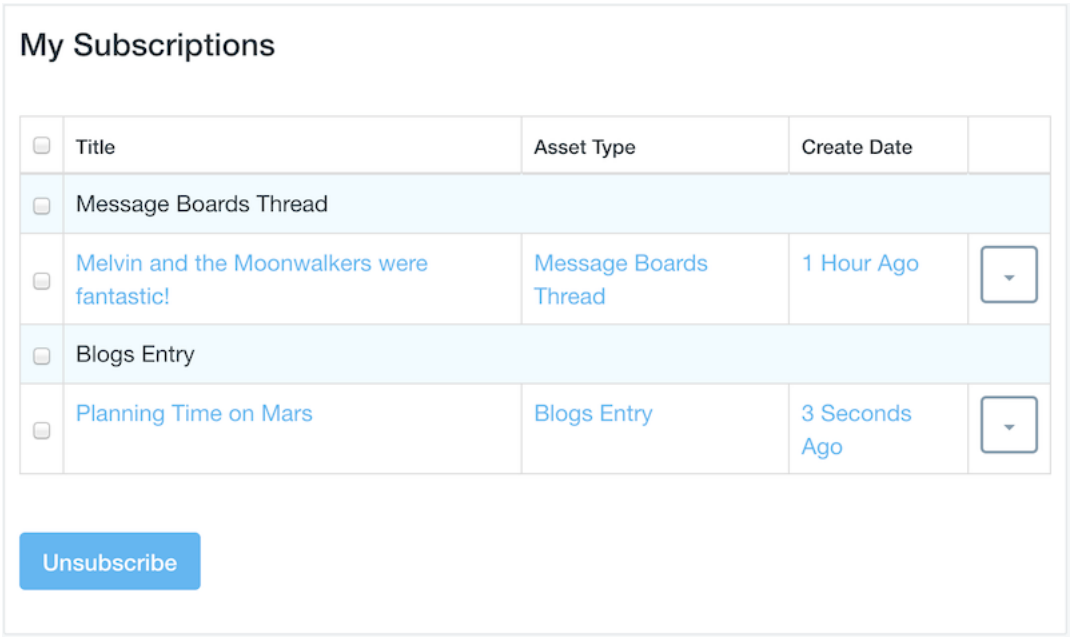


Figure 19.61: My Subscriptions lets you manage all your subscriptions in one app.

Chapter 20

Creating A Social Network

Since the first social networks rose to popularity, concepts such as *Friend* and later *Like*–previously reserved for direct human interaction–have taken on new meaning in an always-online, information driven culture. It could be argued that social networks have transformed the way people interact with their friends, relatives and colleagues. Friends, connections, followers, circles, and lists have enabled people to connect and stay connected in ways they'd never been able to before. Many organizations are now realizing the importance of leveraging social interactions for more than just recreation. Liferay DXP's robust social features make it a great platform for business web sites, casual web sites and everything in between.

Liferay DXP's social applications ship with the product, but they only scratch the surface of what's available on the platform. For example, these applications let you connect your users as Friends, but the API allows for all kinds of relationships, such as coworker, spouse, child, follower, romantic partner, and more. If you want to take advantage of some of these other relationships, your developers can use these apps as a starting point for implementing the full power of Liferay DXP's social platform.

Out of the box, Liferay DXP has a constantly improving set of social features that help you encourage social interactions on your own website as well as leverage the power and popularity of other social networks. In this chapter, you'll learn about

- General use social portlets
- Social portlets for personal pages
- Configuring personal pages for social networking
- How to connect users using Liferay DXP social relations
- Social Activity
- Integrating Liferay DXP with other social sites

When you're finished, you'll be well equipped to use Liferay DXP to power your social network.

20.1 Leveraging Social Applications, Activities Tracking, and User Connections

Liferay DXP has many apps available for social interaction and collaboration. Some of these apps are designed to help you work together as a team, while others are designed to foster social interactions between team members at your organization.

Out-of-the box, Liferay DXP includes these social apps:

- Activities
- Group Statistics
- User Statistics
- Requests

The following social apps are available in Liferay's Social Networking app on Liferay Marketplace:

- Summary
- Wall
- Friends
- Members
- Friends' Activities
- Members' Activities
- Meetups
- Maps

Liferay's Social Networking app is available for both Liferay Portal CE and Liferay DXP:

- Liferay CE Social Networking
- Liferay Social Networking (DXP)

Some of the social networking apps should be used on your site's public pages, while others should be used as part of a user's personal site. As you might expect, the apps intended for a personal page foster simple social interactions, while the ones that can be placed on any site help teams interact and improve productivity.

Unless otherwise noted, these apps have minimal configuration options. Some of them provide additional options for customizing feed lengths or display styles. Additional styling changes can be made through custom CSS.

Liferay DXP's Social Tools in Action

To get started with Liferay DXP's social features, you first want to add social apps to your users' public personal pages. You can set things up any way you want, but for simplicity's sake, this example shows something that's fairly similar to the original Facebook layout. You'll give your users a way to send and receive connection request using their Profile and Dashboard pages.

Setting up Users' Personal Pages Before you start adding apps to pages, you should configure Liferay DXP so that everyone (or some subset of everyone) has the same social features. You can do this in two ways:

User Groups: You can create a user group site by placing users into a group. The pages and apps defined by the user group site are copied to members' personal sites. This lets you control whether users can modify pages, and you can push changes out to users in the future. Once the site template is assigned to a user group, you can make it so all users are members of this group. The advantage of this is that it can be managed entirely through the interface, and it's easy to configure. If you base your user group's site on a template, you can use the *Enable propagation of changes from the site template* option to manage all user pages by changing the template. This is the recommended way to manage personal pages across the portal.

Portal Properties Configuration: The legacy way to do this is with the configuration file. You can specify a default layout and applications for personal pages in your `portal-ext.properties` file. Note that this method applies changes to all users' personal sites. It does not, however, provide as much maintainability or as many customization options as user group sites does. User group sites allow you to choose what's modifiable by

the user. For more information on the `portal-ext.properties` method, search for *Default User Private Layouts* and *Default User Public Layouts* in the properties documentation.

Because it's the recommended method, use the user group method to create the layouts. As an administrator, go to the Control Panel and select *Site Templates* from under the *Sites* section. Click *Add* ( ) and fill out the form. Call your new site template *Social Layout Public*. Click *Save*.

Figure 20.1: You can give your site template a custom name and description and also specify several configuration settings..

Tip: Disabling *Allow Site Administrators to Modify the Pages Associated with this Site Template* only prevents users from modifying the specific pages contained in the template. It does not keep a user from adding or modifying additional pages.

Now repeat the process and create a site template called *Social Layout Private*.

You need to give your users a way to request connections and also to respond to connection requests. To do that, you'll use the Requests application and the Summary application. The Requests application goes on users' dashboard pages, as they are private pages just for that user. The Summary application goes on users' profile pages which are public. This way, users can visit the profile and request a connection.

Once you've created the templates, click the *Go to Other Site* button. Click the link for the Social Layout Public site. Now you want to change the name of the page from the default to *Welcome*. Under Navigation → Pages, edit the default page (labeled *Home*) and change its name to *Welcome*. Click *Save*. Now you'll add an application to the page. Click the *Go to Site* link in the menu and then click the *Add* button ( ). Under *Applications* → *Social*, drag the *Summary* application to the column on the left.

Awesome! You've now set up a site template that contains the Summary application for users' public profiles. Now you need to do the same thing for the Requests application and users' private dashboards. Click the *Go to Other Site* button, but this time click the link for the Social Layout Private site. As you did

before, change the name of the default page from *home* to *Welcome*. Then click the *Go to Site* link, click the *Add* button, and under *Applications → Social*, drag the Requests application and drop it into the leftmost column.

You're almost there. Click *Control Panel* and select *User Groups* from the *Users* section. Once there, click *Add* and name the group *Social Users*. When creating a user group, you can select site templates to apply to those users' profiles and dashboards. For *My Profile*, select *Social Layout Public*. For *My Dashboard*, select *Social Layout Private*. Click *Save*.

Now go to *Control Panel → Configuration → Instance Settings* and select the *Users* section. Go to the *Default User Associations* tab and enter *Social Users* in the User Groups section. Now all users on the portal get a Social Profile page.

Awesome! You've now enabled your users to make social connections. Now the question is, how do we encourage users to visit each others' fancy new profile pages?

Connecting Users Through Collaboration There are many ways that social networks connect users. These generally involve some kind of mutual interest or experience. On a site like Facebook, you can connect with people from school, from work or from other personal connections. On a music based networking site like Last.fm, you can connect with people who have similar tastes to yours. With Liferay DXP's social networking, collaboration is the key to connection.

The Site Members Directory can provide a simple way for users to connect. If you have a site dedicated to Lunar Resort astronauts, you can place a Site Members Directory on that site. Because it lists all the users that have joined that site, users can connect by sending requests to other users on that list. This isn't the worst way to get users connected but it probably won't be very effective. Why not? Well, other than sharing some very basic common interests, users don't interact this way.

The Activities application provides a similar but more effective means of connection. Because it shows a list of what other users are doing, this application helps users discover who is among the most active across the site or the instance, and thus who might be a good connection. This application is covered below.

Probably the most effective way users can connect is by interacting with other users. Every application in the Collaboration category provides information on who is contributing, regardless of how. You can see who is creating a thread in a message board, editing a wiki article, blogging or creating a calendar event. Users can use these to connect based on content–if I find your blog interesting, or if you answer my question on the message board, we can use that as a point to connect as friends to further our interactions. This way, instead of our connection being forced or arbitrary, we've connected based on our direct interaction and share a common interest–just like people did before they had the Internet.

Using Liferay DXP's Social Apps

Liferay DXP's social apps provide a default implementation of Liferay DXP's social API. However, this only scratches the surface of the platform's capability. It's also possible to develop your own implementation of Liferay DXP's social API to use different social relationships. Please refer to the Liferay DXP Developer Tutorials or the Javadocs for information about Liferay DXP's social API.

The core social application in Liferay DXP is Activities. It displays information about user activity on the site where you added the app. User activities tracked include updates to the Documents and Media library, blog posts, message boards posts, wiki pages, and bookmarks. Liferay DXP also tracks information about web content but only displays this information if the logged-in user is a site administrator. This application provides a summary of recent site activity. You can use it on a site's public or private pages to show what site members have been up to or you can use it on the public or private pages of a user's personal site. When added to a personal site, the Activities portlet just shows the activities of a single user.

Note that the app provides links to the assets listed in the feed. The links to the assets won't work, however, unless there's a way to display the assets on the page. For example, suppose that the user Joe Bloggs uploaded

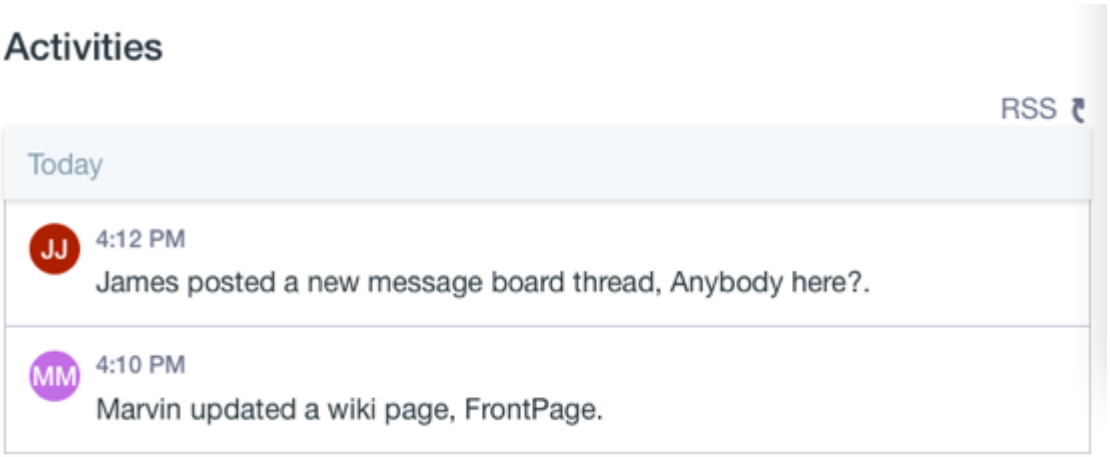


Figure 20.2: The Activities app shows information about asset-related user activity in the current site. It only displays information about web content if the current user is a site administrator.

a document called *Lunar Resort happenings for August* to a site. An Activities app on that site would show a link to the *Lunar Resort happenings for August* document.

In addition to the Activities application, there are several other social applications you can use.

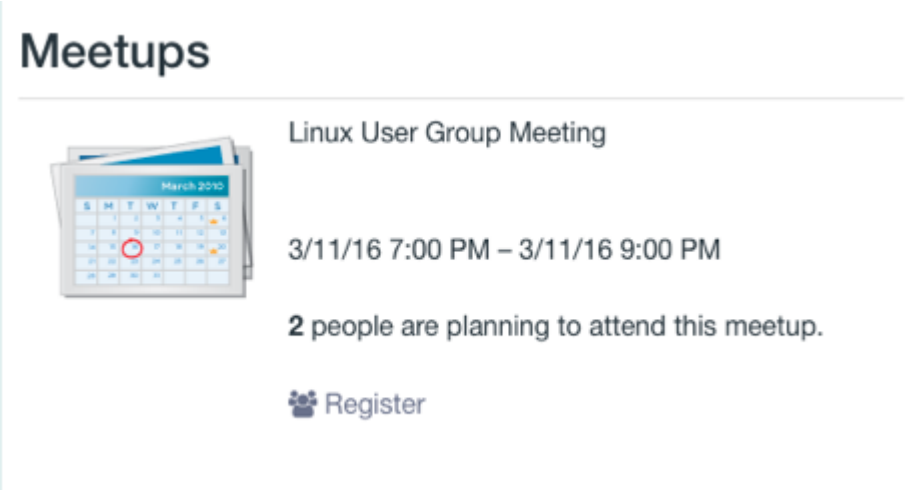


Figure 20.3: Meetups allow users to schedule meetings and hangouts.

The Meetups application is a tool for creating casual meetings for users of your site. Anyone can create a "meetup" and give it a title, description, date/time, maximum number of attendees, price, and provide an image. Meetups are displayed in the application for anyone to view. Users can register for the meetup, which lets the organizer keep track of who's coming.

The options for creating a meetup are essentially the same as those for creating a calendar event.

The Wall application provides a place for users to leave messages on other users' profiles. The messages can only be plain text as no formatting or HTML is supported. Once a post is added to a user's wall, that user can delete it or respond to it with a quick link to post on the original poster's wall.

Friend is only the default social relationship as implemented by Liferay DXP's social portlets. You can design things so that users are automatically connected through Site and Organization membership. And there are many other relationship types beyond Friend: your developers can take advantage of these by using

Liferay DXP's social API.

Now that you have all these social applications running on your system, you might wonder: how can I measure social interaction? How do I identify the best contributors to my site? Liferay DXP has the answer: social activity measurements. This is discussed next.

20.2 Measuring Social Activity

When you have a lot of user interaction on your web site, it can be helpful to try to separate the signal from the noise. Liferay DXP contains a lot of applications which end users can use to communicate with each other and provide information. Some of this information is good and helpful and some of it can be rather unhelpful. Using Liferay DXP's Social Activity feature helps show which users are making real, valuable contributions.

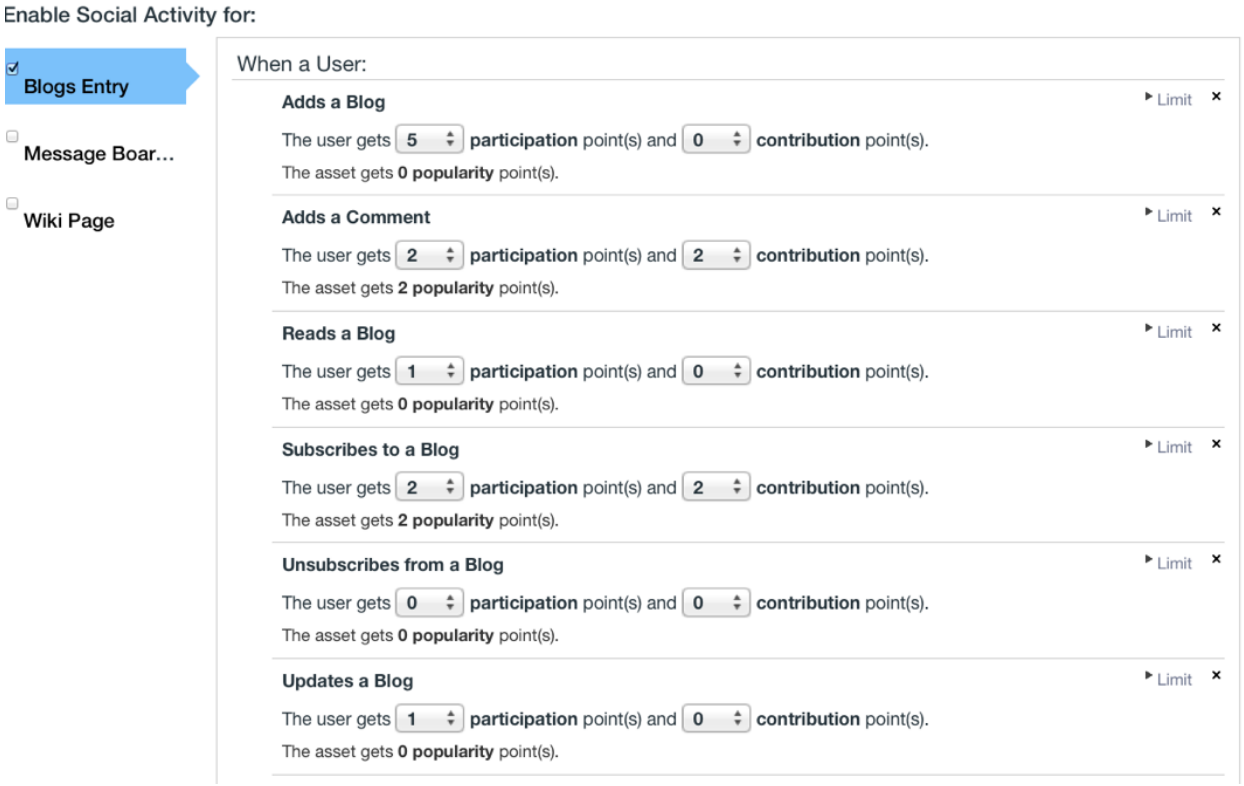


Figure 20.4: The Social Activity page of the Control Panel allows you to enable social activity for assets and specify points for participation and contributions.

To activate Social Activity, you'll first need to determine the collaboration applications you want to use Social Activity to track. There are currently three types of content you can use with Social Activity: Blogs Entries, Message Board Messages, and Wiki Pages. Activation is *a la carte*, so you can use it on one, two, or all three applications. Social Activity tracks three metrics from within each of these applications. Two are for the user (*Participation* and *Contribution*) and the other, *Popularity*, is for the asset involved.

You can find Social Activity in your Site's menu under Configuration. Let's activate Social Activity for Blogs Entries. Check the box next to *Blogs Entry*. You now have options to set point values and limits on several different actions for blogs. You'll notice each item on the list has selectors you can use to set the number of participation and contribution points; popularity points are tied directly to contribution points.

In addition to that, you can expand the box by clicking *Limit* in the top right of each list item. You can use this to set a limit on how many times a user can perform this activity with a specific asset and receive the requisite points. For some activities, you can set limits on both participation and contribution points, but on new content creation, you can only set limits on participation points.

It might not be immediately obvious, but for all actions that do not involve the creation of a new asset, all the contribution points go to the original asset creator and all popularity points go to the original asset. That means if *Votes on a Blog* is set to have 1 *Participation* point and 5 *Contribution* points (and therefore 5 *Popularity* points), the user who votes on the asset will receive 1 participation point, the user who created the asset will receive 5 contribution points, and the asset will receive 5 popularity points.

It's easy to assign points–you can arbitrarily assign points for just about anything–the challenge is making the points significant in some way. As mentioned before, the primary purpose of social activity tracking is to make sure that users who regularly contribute to the portal and participate in discussions are recognized as such. So the central piece of the social activity display is the *User Statistics* application.

The User Statistics application displays a list of users ranked by an amalgamation of their participation and contribution scores. By going to the application's Configuration dialog, you can change some of the specifics of the rankings. There are five check boxes that you can enable or disable:

Rank by Contribution: If this is checked, users' contribution scores are used as a factor in calculating their ranks.

Rank by Participation: If this is checked, users' participation scores are used as a factor in calculating their ranks.

Show Header Text: Determines whether the title shows or only the rankings.

Show Totals: Toggles the display of the users activity score next to their name.

Display Additional Activity Counters: You can toggle the display of any number of other pieces of information next to the user's name in the statistics, ranging from the number of comments on assets a user has created to the number of wiki articles that the user has created. If you want to display multiple data points, you can click the *plus* button to add one and the *minus* button to remove one. You can have as many data points displayed as you want, but displaying too many might make the display a little unwieldy.

The *Group Statistics* portlet provides some more advanced data analytics. If you add it to a page and go to its configuration dialog, you can select the assets to track. You can click the *plus* button to add additional slots and choose from the various metrics available for each slot, covering virtually any action that a user can perform on content in the portal. If you decide that you're displaying too many metrics, you can click the *minus* button for a particular slot on the configuration view to remove it.

There are a wide-ranging number of actions that you can provide social credit for. Users can receive credit for everything from subscribing to a blog to writing wiki articles. You can easily tweak the numbers in the Control Panel if it becomes clear that certain activities are weighted too high or too low.

Social Activity can be a valuable tool for websites that are heavily driven by community-created content. It allows you easily to recognize users who are major contributors, and it indicates to new users whose advice may be most trustworthy. Social Activity is easy to set up and can be configured differently for each site, increasing the flexibility of your website.

20.3 Exporting Liferay DXP Applications To Other Websites

Liferay DXP can publish its applications to other websites in a variety of formats. These range from simple widgets to standards like OpenSocial. This lets you provide your application or content in the context of another website, easily. Read on to find out how this is done.

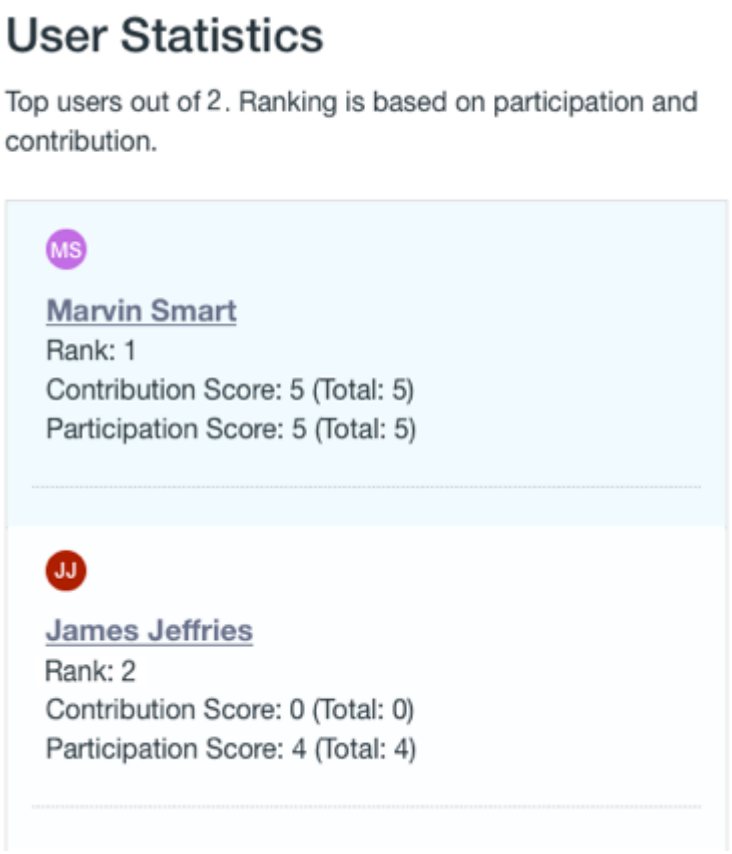


Figure 20.5: The User Statistics portlet gives rankings to promote user contributions and participation.

Sharing OpenSocial Gadgets

OpenSocial consists of a set of APIs for social networking. It may be beneficial for you to share applications from your Liferay DXP server with other sites, such as ighome.com or igoogleportal.com. These sites let users customize their own pages with gadgets. Liferay DXP users can share their applications on any OpenSocial-compatible site. Let's try this now.

For our example, we'll share Liferay DXP's Blogs app:

1. Add the Blogs app to your Liferay DXP page.
2. Click the app's *Options* icon (⋮) and select *Configuration*.
3. In the *Sharing* tab, open the *OpenSocial Gadget* section.
4. Set the selector to *YES* for *Allow users to add Blogs to an OpenSocial platform*. In the *OpenSocial Gadget URL* field, replace `localhost:8080` with the name of your public domain and port.
5. Click *Save*.
6. Close the dialog and click the app's *Options* icon (⋮). There's a new option available named *Add to an OpenSocial Platform*. Select this option to add the app to a page on an OpenSocial platform.

By going through this process, Liferay DXP shared the your app's URL to the OpenSocial platform. The URL you provided is unique to your specific instance of the app. This lets you share multiple instances of the same app as different Gadgets.

You can use this sharing capability to let users view what's happening on your site at a glance.

20.4 Integrating with Facebook, Twitter, and More

Social media is a powerful tool that allows people to communicate and share thoughts and ideas. Integrating social media features into your site is a no brainer. Liferay DXP offers several different ways for you to integrate with social media.

Integrating with Facebook

Facebook is currently the number one social network in the world with somewhere over 1 billion active users. If you're trying to build a community on your site, you don't want to neglect a bridge to that many possible users. With that in mind, Liferay DXP provides a few easy ways for you to integrate your site with Facebook.

Facebook Sign On Like many websites you may visit, any site running on Liferay DXP can be set up to use Facebook for sign in. This makes it easier for users to sign in to your site, since they won't need to remember another user name and password. See the article in the Discover/Deployment section for information on using Facebook to sign into Liferay DXP.

Using Your Application as Facebook Applications You can add any Liferay DXP portlet as an application on Facebook. To do this, you must first get a developer key. A link for doing this is provided to you in the Facebook tab in any application's Configuration screen. You will have to create the application on Facebook and get the key and canvas page URL from Facebook. Once you've done this, you can copy and paste their values into the Facebook tab. Once you do that, your application is available on Facebook.

This integration enables you to make things like Message Boards, Calendars, Wikis, and other content on your website available to a much larger audience (unless you already have a billion users on your site, in which case, kudos to you). If you're a developer, you could implement your game or other application on Liferay DXP and then publish it to Facebook.

Integrating with Twitter

Liferay DXP also provides integration with Twitter via the Twitter app on Liferay Marketplace. Place the Twitter app on your private dashboard page. Once you register your Twitter username with the app, your tweets are shown in your Activities application along with your activity feed. Make sure that you get the appropriate version of the app for your Liferay DXP instance.

There are two ways to register your Twitter username with the app after you install it. The first is through the Twitter portlet after you add it to your dashboard page. Click the link in the portlet and then enter your Twitter username in the Twitter field listed in the *Identification* tab under the *Social Network* heading. The second way to register your username is by accessing your account settings directly and following the same steps.

It's important to note that your tweets are displayed only in an Activities application. To view your tweets, add the Activities application to your user public profile page. Users can then view your profile to see your tweets.

The Twitter feed is updated every five minutes by default, so you need to wait a few minutes after registering your Twitter username for your tweets to show up. This value can be changed through System

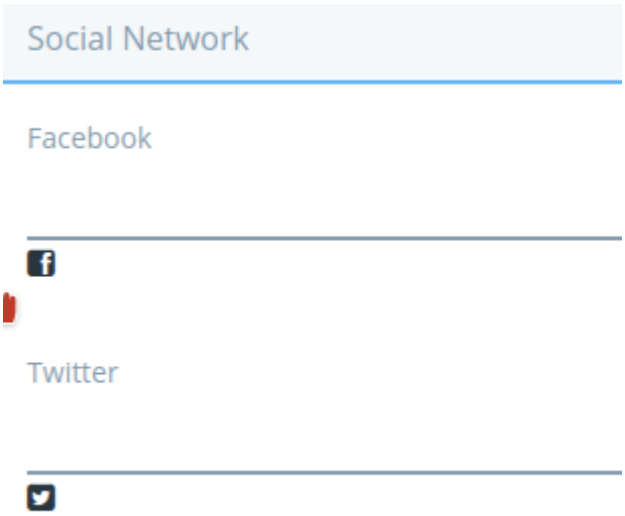


Figure 20.6: Register your Twitter account.

Settings. Go to *Control Panel → System Settings*, select the *Other* tab, and click *Twitter Service*. The default value for the *Twitter synchronization interval* is *5* (5 minutes). Set this value to *1* to refresh the Twitter feed every minute.

Keep in mind that your tweets are shown in order according to the date and time that you originally made them. Therefore, if you haven't tweeted in a while, then your tweets might be pushed off the Activities feed by more recent activity. You can change the number of items shown in the Activities feed in its Configuration.

Using Social Bookmarks

So far, you've learned how to integrate your site with Facebook, but what about the other social networks? There's an app for that: Social Bookmarks. When this feature is enabled, icons are displayed in a menu or below the content that allows users to share their content to Facebook, Twitter, and Google+. This feature is enabled by default in the Blogs portlet, and can be enabled for other asset-enabled portlets through the *configuration* menu of the portlet or in System Settings.

If you want more social bookmarks than come out of the box, be sure to install the Social Bookmarks app from the Marketplace (available for CE and EE). The Social Bookmarks app adds the following bookmarks to the default set of social bookmarks:

- AddThis (containing 199+ social media sites)
- Compartir
- Delicious
- Digg
- Evernote
- LinkedIn
- Newsvine
- Reddit
- Slashdot
- StumbleUpon
- Tuenti

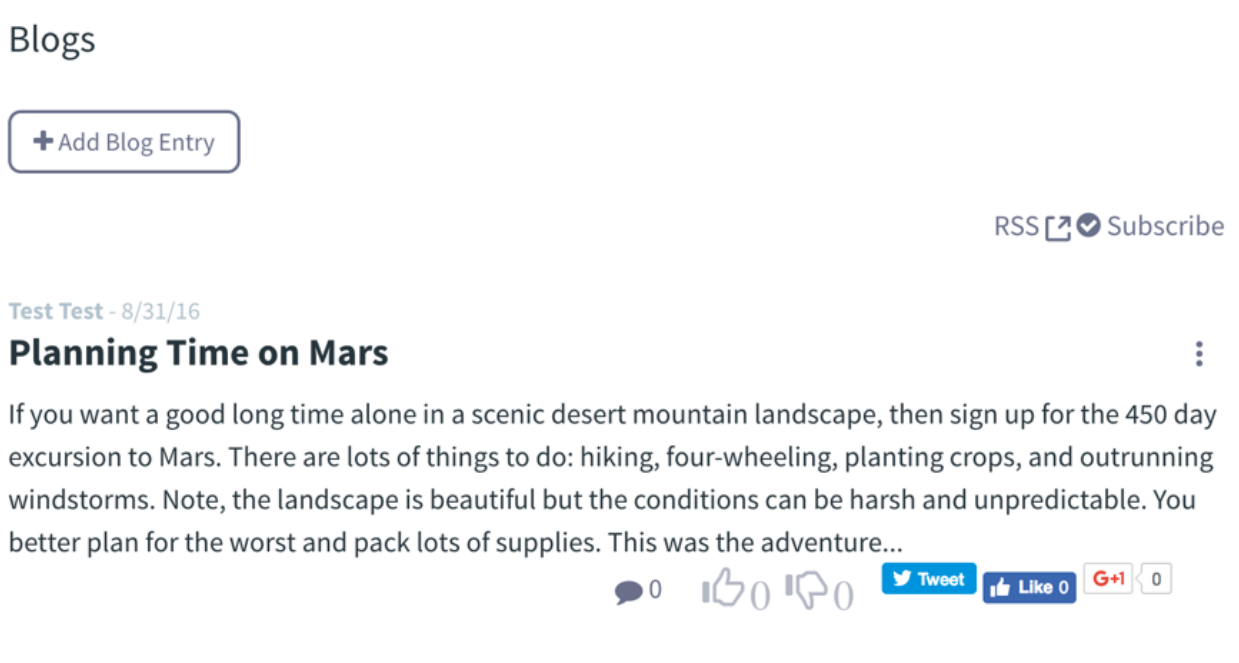


Figure 20.7: Social Bookmarks are enabled by default in the Blogs portlet.

Note: The Blinklist and Technorati bookmark links are no longer active, and have therefore been removed from the Social Bookmarks app.

Chapter 21

Search

Sites built on Liferay DXP often feature lots of content split over lots of asset types. Web content articles, documents and media files, and blogs entries are just a few examples. Most content types in Liferay DXP are *assets*. Under the hood, assets use the Asset API and have an Indexer class. Any content that has these features can be searched in the Liferay DXP Search application.

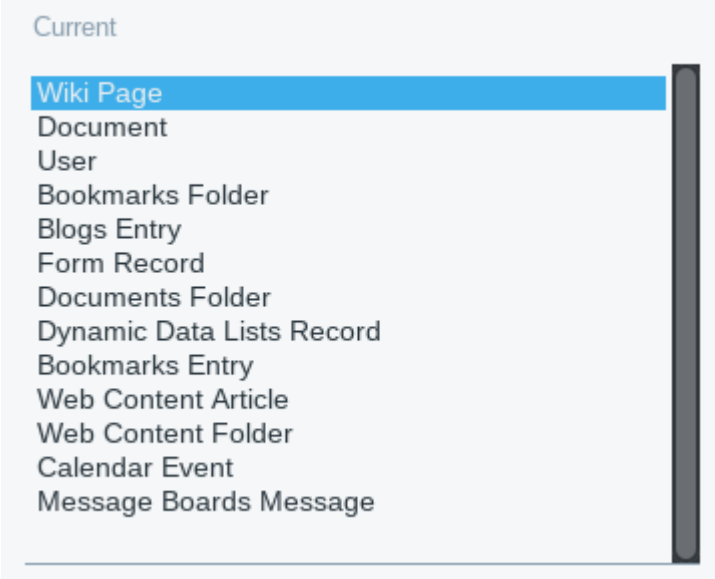


Figure 21.1: There are many searchable out-of-the-box asset types.

21.1 Searching the Index, not the Database

Liferay DXP stores its data in a database. You might incorrectly assume that you're directly searching the database when you use Liferay DXP's Search application. Liferay DXP instead leverages a search engine for its search capabilities. Using a search engine like Elasticsearch lets you convert searchable entities into *documents*. Documents are created and added to the *index* at the same time they're added to the database. They're also updated whenever the database is updated, and deleted from the index when the backing

entity is deleted from the database. When you enter a search term in Liferay DXP, nothing happens in the database, because the documents are already indexed on the Elasticsearch server, and that's where the search is executed.

It's worth adding the complexity of a search engine, rather than searching the database directly, for performance reasons and for some of the features that search engines provide, like algorithms that give you the ability to use relevancy scores. For more technical details, see the Introduction to Search developer article.

21.2 Leveraging Elasticsearch in Liferay DXP

The default search engine used by Liferay DXP is Elasticsearch, which is backed by the Lucene search library. There's an Elasticsearch server embedded in Liferay DXP bundles, which is handy for testing and development purposes. Production environments must install a separate, remote Elasticsearch server (or even better, cluster of servers). For information on how to set up Elasticsearch for Liferay DXP in production, read the deployment guide.

21.3 Liferay DXP Search Features

Searching is simple and straightforward. Find a search portlet (there's one embedded in every page by default), enter a term in its search bar, and click *Enter*.

Figure 21.2: There's a search bar embedded on all Liferay DXP pages by default.

A results page is displayed. If there are hits to search engine documents, you'll see them as search results in the right hand column. In the left hand column you'll see search facets.

The search bar, search results, and search facets make up three powerful features in Liferay DXP's search UI.

Search Bar

The search bar is simple: it's where you enter *search terms*. Search terms are the text you send to the search engine to match against the documents in the index. The documents that are returned are where this gets interesting.

Search Results and Relevance

The search term is processed by an algorithm in the search engine, and search results are returned to users in order of relevance. Relevance is determined by a document's *score*, generated against the search query. The higher the score, the more relevant a document is considered. The particular relevance algorithm used is dependent on the algorithms are provided by the search engine (Elasticsearch by default).

In short, answers to questions like those below determine the relevance score of a hit (matching document):

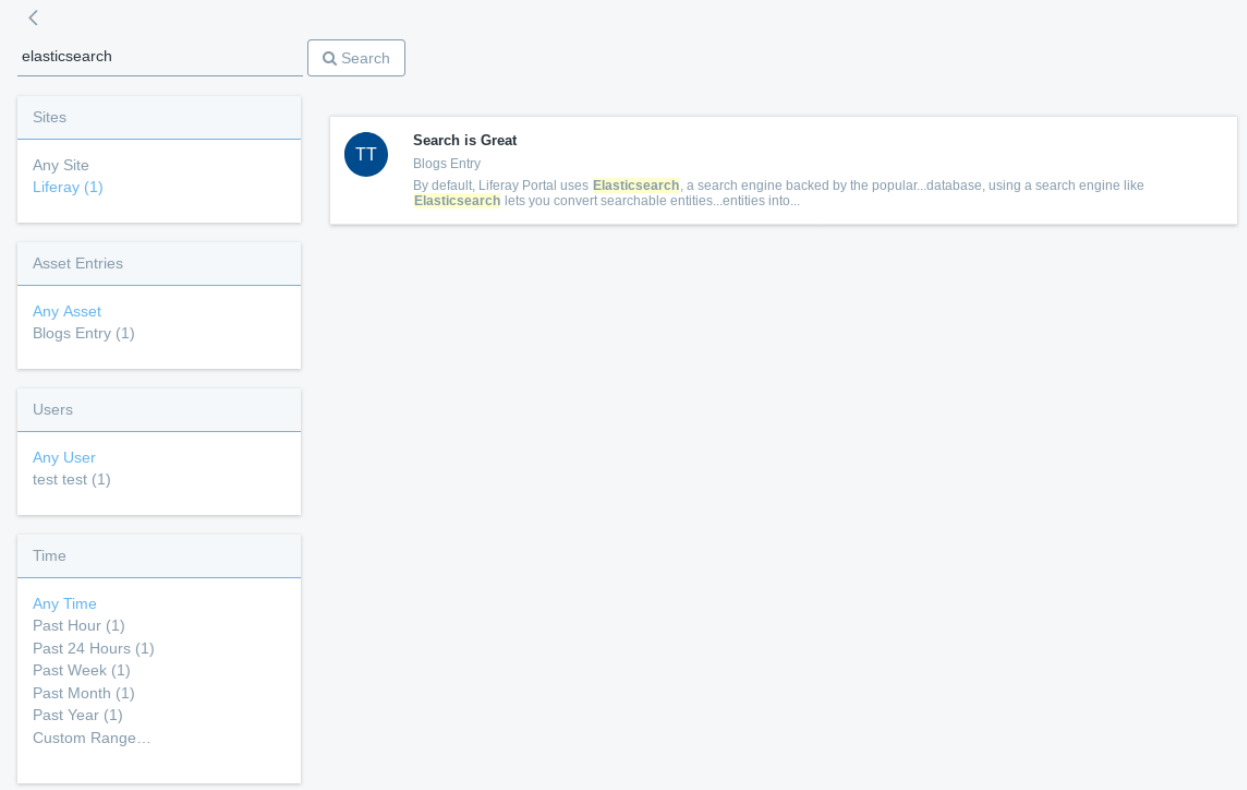


Figure 21.3: There's a search bar embedded on all Liferay DXP pages by default.

- How many times does the search term appear in a document's field?
- How many times does the search term appear in the same field of all the other documents in the index?
- How long is the field where the term appears?

If the search term appears with greater frequency in the field of one document than is the case for the same field in other documents, the score will be higher. However, if it's a long field (like a *content* field for a Blogs Entry document) then the presence of the search term is discounted. Its presence in a shorter field (like a *title* field) produces a higher relevance score. See the Search Results article for a longer discussion of relevance.

Search Facets

Facets are a core feature of the @prodcut@ Search application.

Facets allow users of the Search application to filter search results. Think of facets as buckets that hold similar search results. You might want to see the results in all the buckets, but after scanning the results, you might decide that the results of just one bucket better represent what you're looking for. So what facets are included in Liferay DXP by default?

- Site
- Asset type
- Asset tag
- Asset category

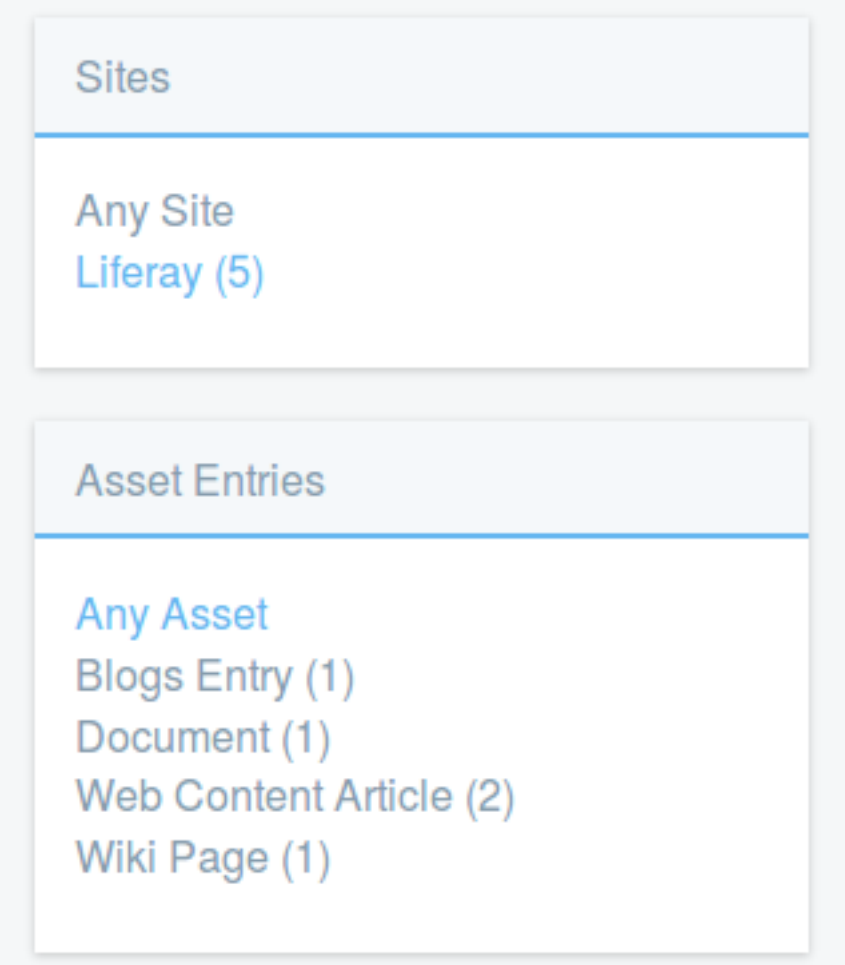


Figure 21.4: *Sites* and *Asset Entries* are two of the facet sets you'll encounter. They let you drill down to results that contain the search terms you entered.

- Folder
- User
- Modified date

You've probably used something similar on any number of sites, especially with online commerce. You search for an item, are presented with a list of results, and a list of buckets you can click to further refine the search results, without entering additional search terms. Search facets work the same way in Liferay DXP. Facets are, of course, configurable.

21.4 Searching for Assets

As explained in the Search introduction, all indexed assets can be returned as search results. Since any developer can create their own assets, your Liferay DXP instance might have additional asset types beyond the ones that Liferay DXP ships with by default.

Searching for Users: When you click an asset in the search results, it's displayed in an Asset Publisher (unless you use the *View in Context* option for the asset). Users are different, though. Think of them as invisible assets, not intended for display in the Asset Publisher application. While users appear as search results with

other indexed assets, when you click one you're taken to the user's profile page. If public personal pages have been disabled, clicking on a user from the list of search results shows you a blank page.

Versioning Note: If there are multiple versions of the same content, the latest approved version is the only one searched.

Search Bar

The search bar is where users enter the search context. Users enter search terms, hit their *Enter* button (or click the magnifying glass icon), and they're taken to a maximized view of the search portlet displaying any results and facets that apply.

Figure 21.5: The Search application displays just the search bar in its default view.

Entering Search Terms Liferay DXP, backed by Elasticsearch, supports *full text search*. It does not, however, support advanced search syntax, such as Elasticsearch's query string syntax. That means you can't search specific fields, use wildcards like *, or boolean operators like *AND/OR/NOT*. Liferay is weighing the costs and benefits of enabling the use of string queries out of the box in a future version of Liferay DXP.

Prefix Searching *Prefix* searching is cool. If you're searching in a site for classical musicians, you might search for the term *instrument*. This search of course returns documents with the full word in them, but it also returns variants with *instruments* as the prefix. For example, results with *instruments*, *instrumental*, and *instrumentation* would also be returned.

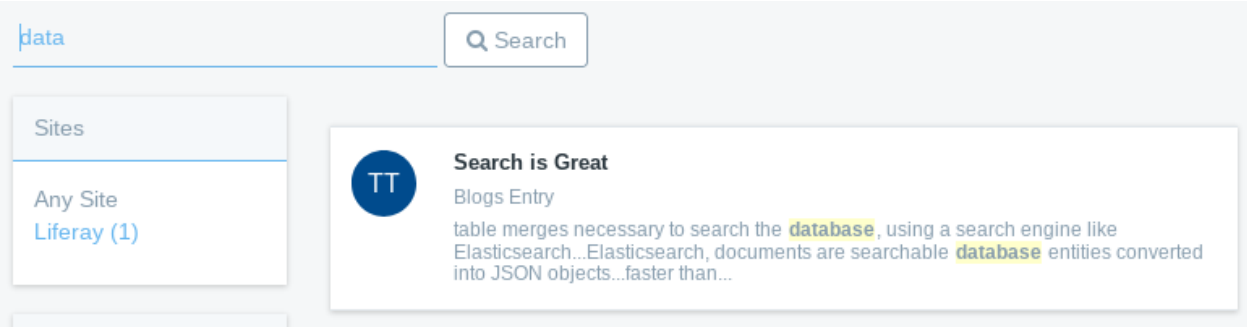


Figure 21.6: Searching for *data* in the Search application also returns database.

Another way to ensure users see results is using the spell check settings.

Spell Checking User Queries

Spell check settings allow administrators to configure the Search application so that if a user types a search term that doesn't return many results (for example, a slightly misspelled werd), the user can be prompted to improve their search.

To configure the spell check settings, click the options ( ) button and select *Configuration*. The tab displayed is *Display Settings*, and next to it is the Spell Check Settings tab. Here you can configure the Search application to help users when their initial query doesn't yield many results.

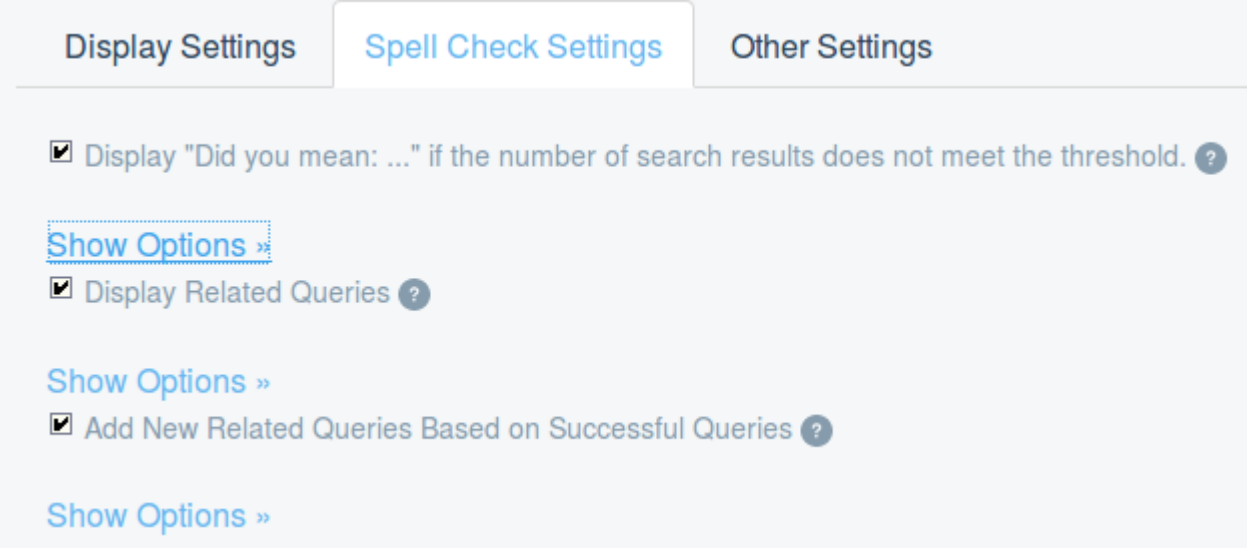


Figure 21.7: Configure the spell check settings to allow for user input mistakes and help lead users to results.

There are three main settings here:

Display "Did you mean..." if the number of search results does not meet the threshold. : Present users alternate, spell checked search queries if their search did not return a minimum number of results (50 by default).

Display Related Queries Display up to a maximum number of alternative queries (5 by default) if the user's query doesn't return a minimum number of results (0 by default).

Add New Related Queries Based on Successful Queries Index a user's search query if it produces a minimum number of results (50 by default). If the Display Related Queries setting is enabled, it can be used as a related query for similar search queries that don't produce enough results.

Note: To enable the spell checking behavior described above, you must first trigger a reindex of the spell check indexes. Navigate to the *Control Panel* → *Configuration* → *Server Administration*. Find the Index Actions entry for *Reindex all spell check indexes*. Click *Execute*, and look for a message in your log your indicating that the spell checking indexes were created successfully:

```
16:19:35,793 INFO  [liferay/search_writer/SYSTEM_ENGINE-24][BaseSpellCheckIndexWriter:257]
    Start indexing dictionary for com/liferay/portal/search/dependencies/spellchecker/en_US.txt
16:19:41,932 INFO  [liferay/search_writer/SYSTEM_ENGINE-24][BaseSpellCheckIndexWriter:2910]
    Finished indexing dictionary for com/liferay/portal/search/dependencies/spellchecker/en_US.txt
```

Once the spell checking indexes are created, your spell check settings become effective.

In addition to enabling or disabling the spell check settings, thresholds for all of the settings are changeable, to produce the desired behavior.

21.5 Search Results

The ideal search experience involves a user entering a search term, waiting an infinitesimal amount of time, and having the perfectly matching asset delivered to them at the top of a list of other extremely relevant hits. Like this:

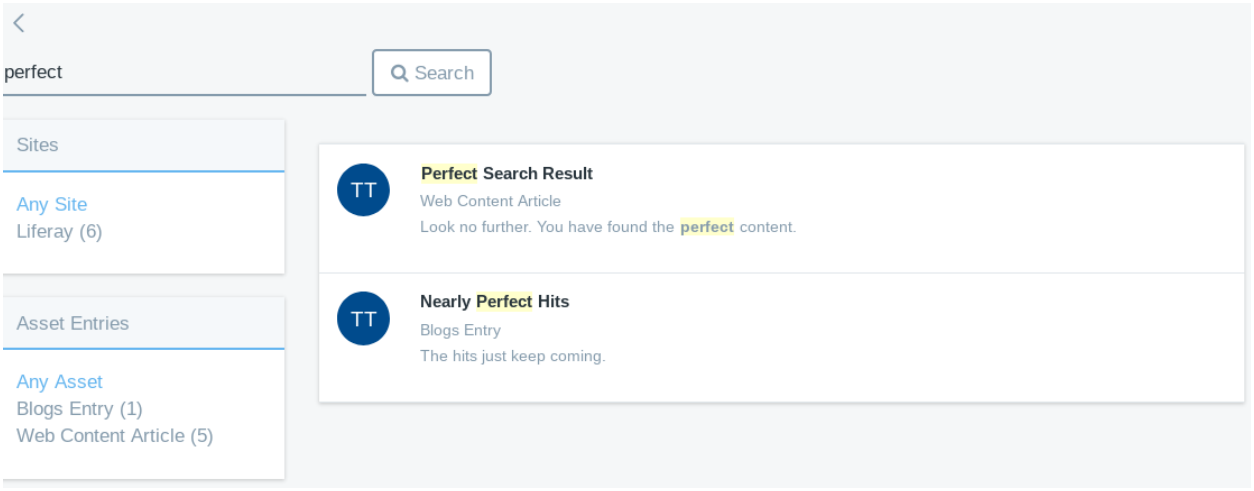


Figure 21.8: The goal is to return the perfect results to users searching your site.

The developers of each asset control much about how the asset's information is stored in the search engine (this process is called *indexing*), and how its information is searched and returned in the search results. Developers who dislike how a particular asset behaves in search can use an *Indexer Post Processor* to modify the asset's indexing behavior, and how search queries are constructed to look up the assets in Liferay DXP.

There are also ways to influence the way search results are displayed from the user interface. This article covers the following topics:

- Configuring the display of search results
- Understanding search results relevance
- Filtering search results with facets
- The effect of permissions on search results
- Search results in the staging environment
- Search results summaries
- Search term highlighting

Search results, called *hits* in the backend search infrastructure, are the end of the road, the destination, to some users. To others, they're just the beginning of the journey. Either way, you can configure how they're displayed.

Configuring Results Display

The search application has a nice format for displaying search results. It also allows users to click on a specific result so they can look at it in more detail. Configure the application's display options by clicking its options menu ( ) and selecting *Configuration*. The tab displayed is *Display Settings*.

The Scope setting is really important. By default, searching is done on *This Site*, which means only the assets associated with the site where the search is executed. To expand the scope of the search to the entire Liferay DXP instance, select *Everything*. To let the user choose which scope they want to search, select *Let the User Choose*.

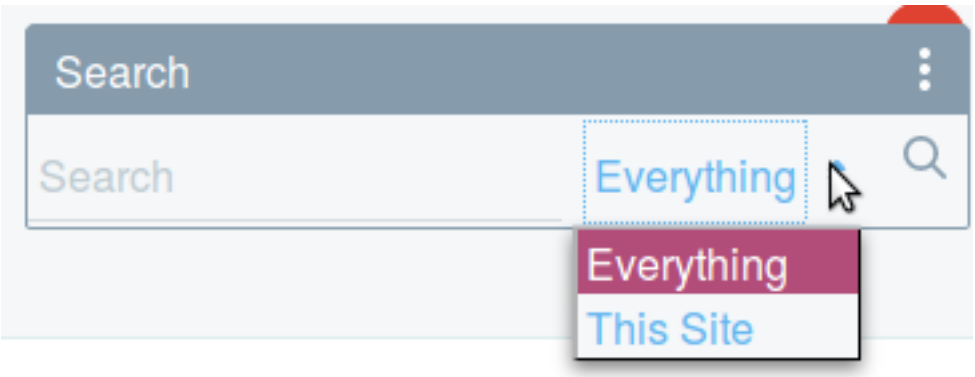


Figure 21.9: The *Let the User Choose* scope option enables a drop-down menu in the search bar where users can set the scope of their search.

The list of facet settings on this page is also quite important. Learn more about facets and their configuration options in a separate article.

For more display options, click the *Other Settings* tab. There are several options:

Display Results in Document Form Display results as search documents. Never use this in production. Developers use this feature to view search responses in their indexed, document-based format. Part of a developer's job when writing search indexers is to convert documents (the objects that get indexed) to the actual object and back again. This option allows developers to see how their objects are being indexed. Once enabled, expand each individual result whose document form you'd like to view by clicking the *Details...* link below the result summary.

View in Context When an asset is clicked, show it in its native application. For example, if you click on a blog post in the search results, you'll be taken to the page where the blog entry is posted in the Blogs application. Note that you will no longer be in the search context after clicking on a search result. When this option is unchecked, the asset displays in an Asset Publisher window while still in the search context. If you have the right permissions, you can even edit the content directly from the Search context. Click the back arrow to return to the search results.

Display Main Query Show the exact search query that the app generated to the search engine. Never use this in production; this is for development purposes only.

Display Open Search Results Show results from third party Open Search plugins, if they are installed. This is for backward compatibility only: developers are encouraged to re-design their search code, and then custom assets are aggregated with native Liferay DXP assets seamlessly.

For further reading, check out how to return suggestions for better search terms (for example, "Did you mean...") when not enough results are returned initially.

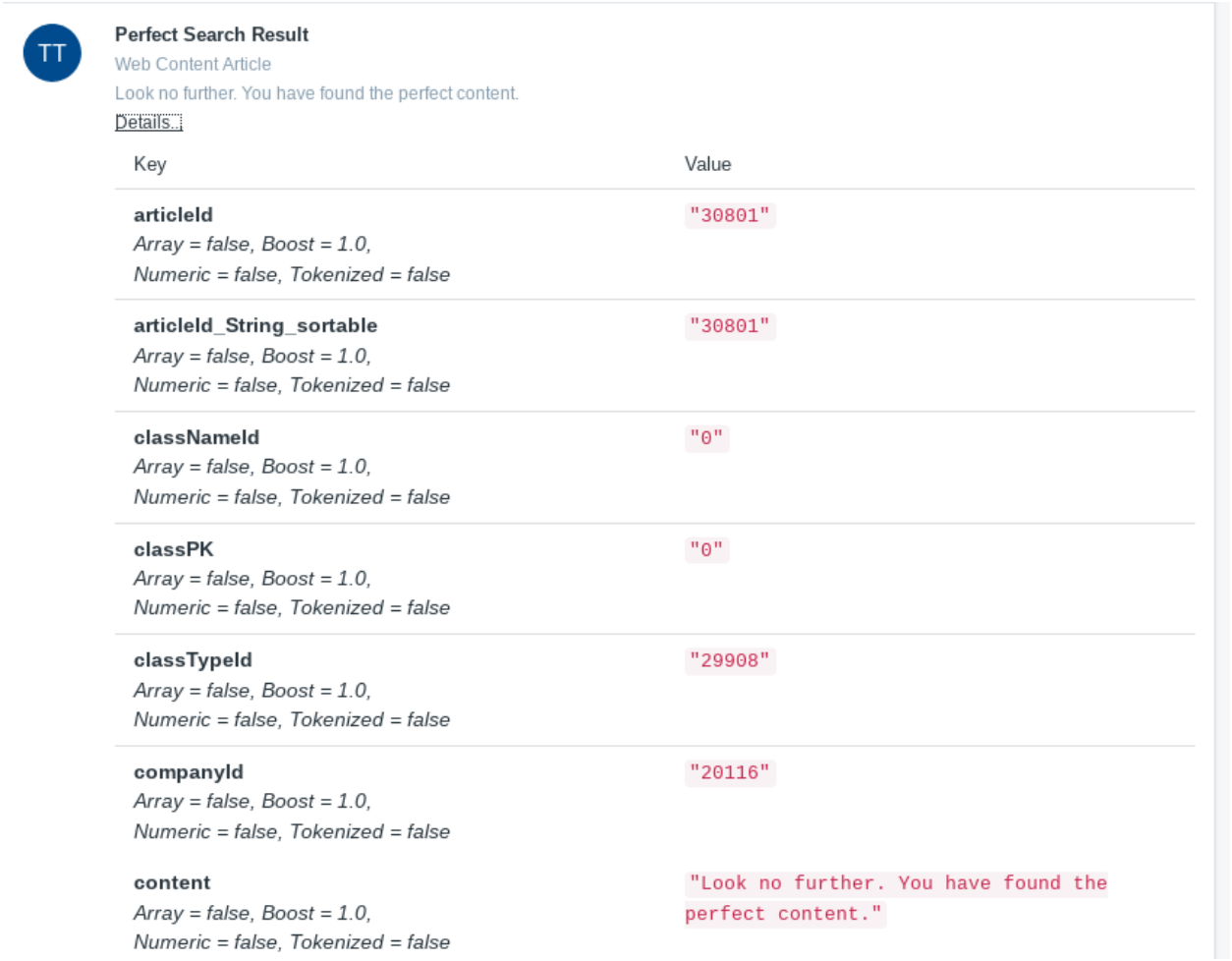


Figure 21.10: Viewing results in their document form shows exactly what's being indexed for a particular asset.

Filtering Results with Facets

Results are filtered using *facets*. The usage by end users is quite simple and intuitive. Most users will have encountered similar filtering capabilities in other online applications, particularly during online commerce activities. Users enter a search term, are presented with a list of results and search facets, which you can think of as buckets that group results together if they share a common characteristic.

Administrators can configure facets. Read about configuring facets to learn more.

Search Results Relevance

How does the search engine decide which results to return at the top of the list? It uses the concept of *relevance*. Relevance is based on a score calculated by the search engine. There are numerous factors contributing to the total score of a returned document. This section aims to give an overview and provide general understanding on the calculation of relevance.

The relevance scoring approach used in Liferay DXP can be distilled into three principles:

1. Term Frequency: If a term appears more than once in the fields of a document, the document's relevancy score is higher than if it only appeared once. Recall that the document is the entity being

searched for in the search engine. It's like the corresponding entity in the database, but may not include all of the same fields. Example:

2. Inverse Document Frequency: Matching terms that are rare in the index provide a higher relevance score than those that are more common. Example:

3. Field-Length Norm: Matches from shorter fields, like title, score higher than those in longer fields, like content.

Those principles determine the order of results returned in the search portlet. To look in depth at the relative contribution of each to a result set's documents, access Elasticsearch's API via URL, like this generalized form:

```
http://host:port/index-name/type/_search?q=title:searchTerm&explain
```

Consider a specific example for an Elasticsearch running on `localhost:9200`, with an index name of `liferay-20116`, with a type of `LiferayDocumentType`, and searching the title field for the word *ziti*. Importantly, the `explain` option is appended to the URL, ensuring that the scoring details are returned for each result:

```
http://localhost:9200/liferay-20116/LiferayDocumentType/_search?q=title:ziti&explain
```

The results are returned in JSON format:

A logical outcome of these three scoring principles is that *title is king*. A match in the title field will produce a good score, because of the field length norm principle. Title fields are usually short, so a match there will score well. A phrase match in the title field, where multiple consecutive words from the search term match a phrase in the title field, is even better. An exact match between the search term and the title is pretty much the supreme overlord of search scoring.

See the Elasticsearch documentation for more information on relevancy scoring.

Permissions and Search Results

Liferay DXP includes a robust role-based permissions system. It's important that users lacking permission to view an asset also can't see it in the search results. A logged in user with the Site Administrator role will likely see more search results than a guest user to the site. To understand whether a user has permissions to see an asset in the search results, the answer to this question must be *yes*:

Does the user occupy a role that has VIEW permission on an asset?

If the user has VIEW permission on an asset, then the asset is returned in the search results.

Another important concept is that of *post filtering*. Search results are returned from the search index to the Search portlet, and a final round of permission checks is performed prior to presenting results on the UI. For example, the user searches for the term *liferay*, and the search engine returns all relevant forum posts. As the Search Portlet iterates thru the list of relevant forum posts, it performs one last permission check of the post to ensure the user can view the post and the categories it exists within. If a matching forum post exists in a category the user doesn't have permission to view, it isn't displayed in the list of search results. Post filtering is done prior to display, so the facet count is not updated. This can result in inaccurate result counts in the Search Portlet. This behavior will be fixed in the next version of the Search application.

Figure 21.11: The scoring explanation of search results, displayed in JSON.

Search and Staging

Liferay DXP supports the concept of staging, where content is first placed in a preview and testing environment before being published for consumption by end users (on the live site). Content added to the search index is marked so that the search API can decipher whether an item is live or not. In the live version of the site, it's quite simple: only content that's marked for the live site is searchable.

In the staged version of the site, all content, whether live or staged, is searchable.

Result Summaries

Search results must be displayed to users to be useful. If each result was displayed in its JSON document form, users would faint and User Experience Designers around the world might spontaneously combust. Liferay values end users and User Experience Designers alike, so a list of result summaries is returned instead.

So what's included in a result summary? The information from a document that the asset's developer felt is most useful to end users searching for the asset. That means that each asset can have different fields

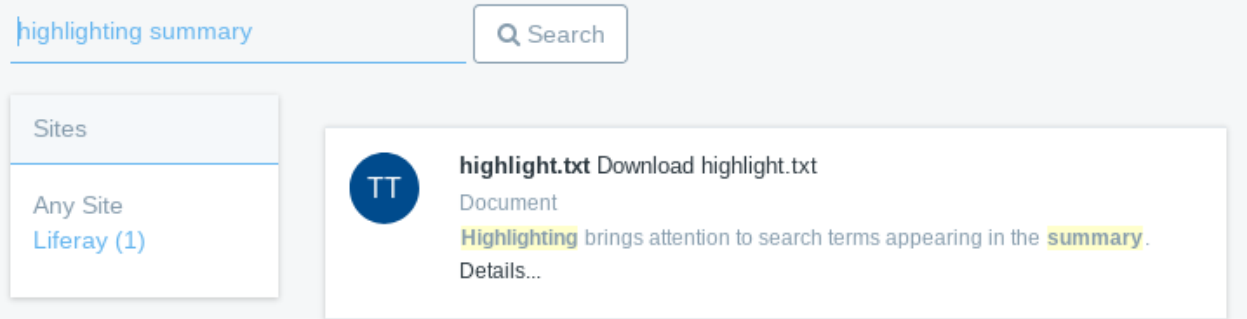


Figure 21.12: Highlighting is useful for drawing attention to your search terms where they appear in the result summary.

included in their search result summaries. For assets with text content, a common summary format is to include the title and the content of the asset. The title is displayed first. The asset type (for example, Document in the example image above) is always displayed on the second line, and a snippet of the content that includes a match to the search term on the last line. Some assets, like Documents and Media documents, display the description field if no content is available to display.

Users are different. Only the user's full name and the asset type (User) are displayed in user result summaries.

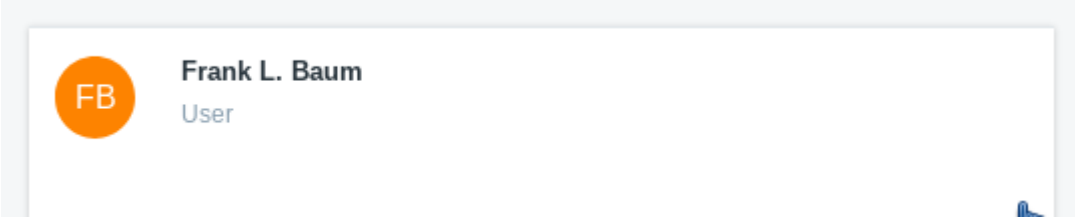


Figure 21.13: User summaries contain only the full name.

For assets that contain other assets (Web Content, Documents & Media, and Bookmarks folders) or whose content is not amenable to display (Dynamic Data List Records and Calendar Events), it makes more sense to display the title, asset type, and description in results summaries. There'd never be anything in a content field for these assets.

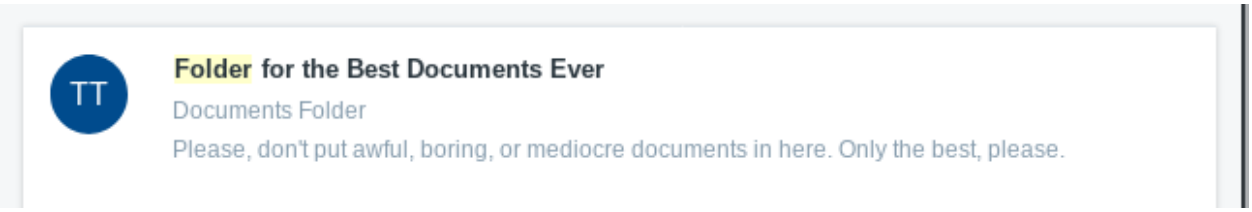


Figure 21.14: Documents and Media, Web Content, and Bookmarks folders include titles and descriptions in their summaries.

Bookmarks entries show the title and the URL.

Highlighting

By now you've probably noticed that search terms appearing in the summary are highlighted.

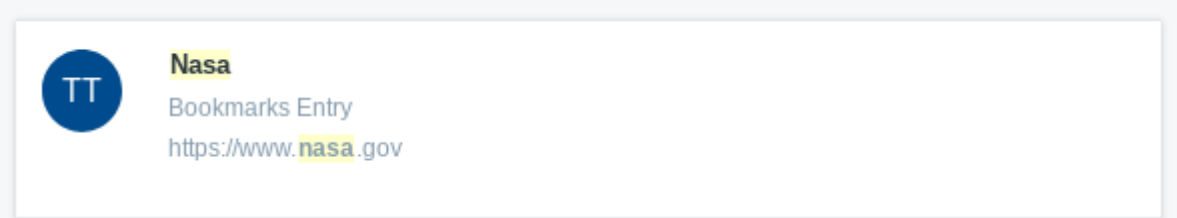


Figure 21.15: Bookmarks Entries summaries show the title and the URL.

Figure 21.16: Some document summaries have lots of highlights, if the search term matches text that appears in the summary.

Highlighting is a helpful visual cue that hints at why the result is returned, but beware. A hit can score well, and thus be returned near the top of the results, without having any highlights. That's because not all indexed fields appear in the summary. Consider a user named Arthur C. Clarke. He has an email address of *acc@authors.org*, which is searchable. Because results summaries for users only contain the full name of the user, searching for Mr. Clarke by his email address returns the user, but no term is highlighted.

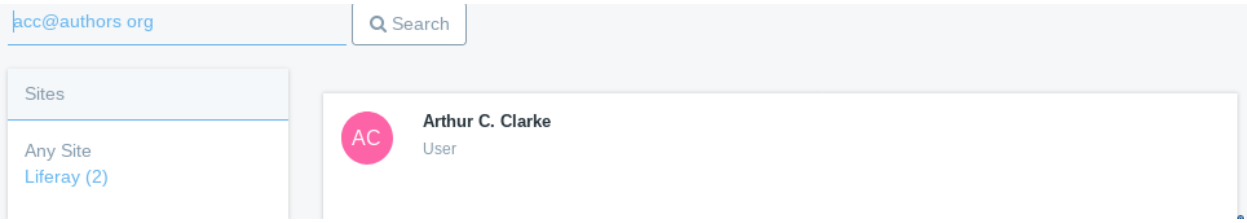


Figure 21.17: : Results that match the search term won't always have highlights.

There are additional cases where search results won't have highlighting, so don't automatically assume the Search application is revolting if you see results summaries with no highlighted terms in them. On the other hand, if the search results list returns a list of results that display only "I'm sorry, [Your Name], I'm afraid I can't do that"1, then the Search application is definitely revolting. Kill your Liferay DXP instance immediately and/or hide under your desk until the AI revolution is thwarted or completed.

1 This is a nod to HAL 9000, supercomputer of *2001: A Space Odyssey* fame.

21.6 Configuring Facets

To get started using faceted search, enter a search term in the Search application's search bar. You'll see a page with results on the right and a collection of *facets*, with the number of search results for each facet (in parentheses) on the left.

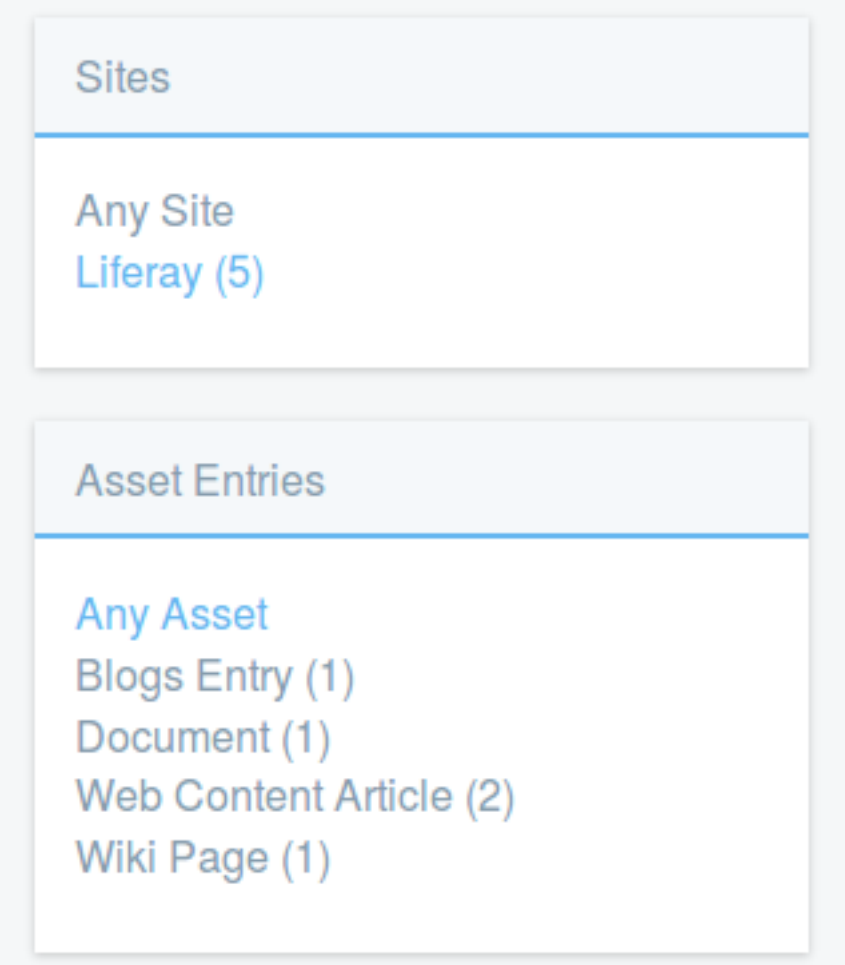


Figure 21.18: *Sites* and *Asset Entries* are two of the facet sets you'll encounter. They let you drill down to results that contain the search terms you entered.

Facets allow users to filter search results by some common characteristic. Think of facets as buckets that hold search results that share something in common. You might want to see the results from all the buckets (the master results list you're first presented), but after scanning the results, you might decide the results from just one of the buckets better represent what you're looking for (for example, all the results from a particular site, or all the results that are Blogs Entries). So what facets are included in Liferay DXP by default?

- Site
- Asset type
- Asset tag
- Asset category
- Folder
- User
- Modified date

In this article, explore how facets are used by end users, and learn how administrators can configure facets for their use case.

Using Facets

After a search is executed, clicking on a specific site from the Site facet filters the search results to only display assets within the specified site. Clicking on a specific user filters the search results to only display assets added by the specified user. The frequency with which the term was found for each facet is listed in parentheses after the facet. Sometimes, viewing assets with certain tags or categories is helpful.

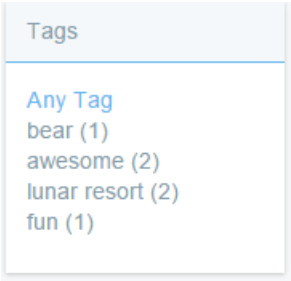


Figure 21.19: Asset tag facets let you see how many assets contain the terms for which you searched *and* contain certain tags. Click on a specific tag to see only content to which the tag has been applied.

Example: Pretend you're an accomplished oboe player (maybe you really are, but if not, pretend), and you're visiting a site for classical musicians. You remember reading a great technical analysis of Johann Bach's compositions, but you forgot to bookmark it (or would it be a *bachmark*?). You enter the term *bach* into the search bar, and, because Johann Bach was a very important and famous composer, you get lots of results: too many, in fact. At first you're discouraged but you remember that there's a site member who produces most of the site's good technical content, who's named *back2bach*. You see that his name is listed in the User facet, and there aren't many results in the facet count (the number in parentheses next to the facet). You click into the facet and quickly find the content you were looking for.

Clicking on a facet narrows down the search results. It's added to the filter list and the results list is refined by the selected facets. If you need to narrow the results further, click another facet (do this as much as you want to drill down into the search results). To remove any of the facets from the filtering of results, click the *Any...* link for the facet type. The following two figures illustrate how this works.

The default facet behavior is quite useful, but you can configure the facets to your liking.

Facet Settings

To configure the Search application's facets, click the options menu ( ) and select *Configuration*. The tab displayed by default is *Display Settings*, and after the Scope setting, the Facets are listed with a Configure link you can click to expand the list of configurations for a given facet.

Facets are configurable to a considerable degree. The following configuration options are available:

Display Facet: Specifies whether the facet appears in search results.

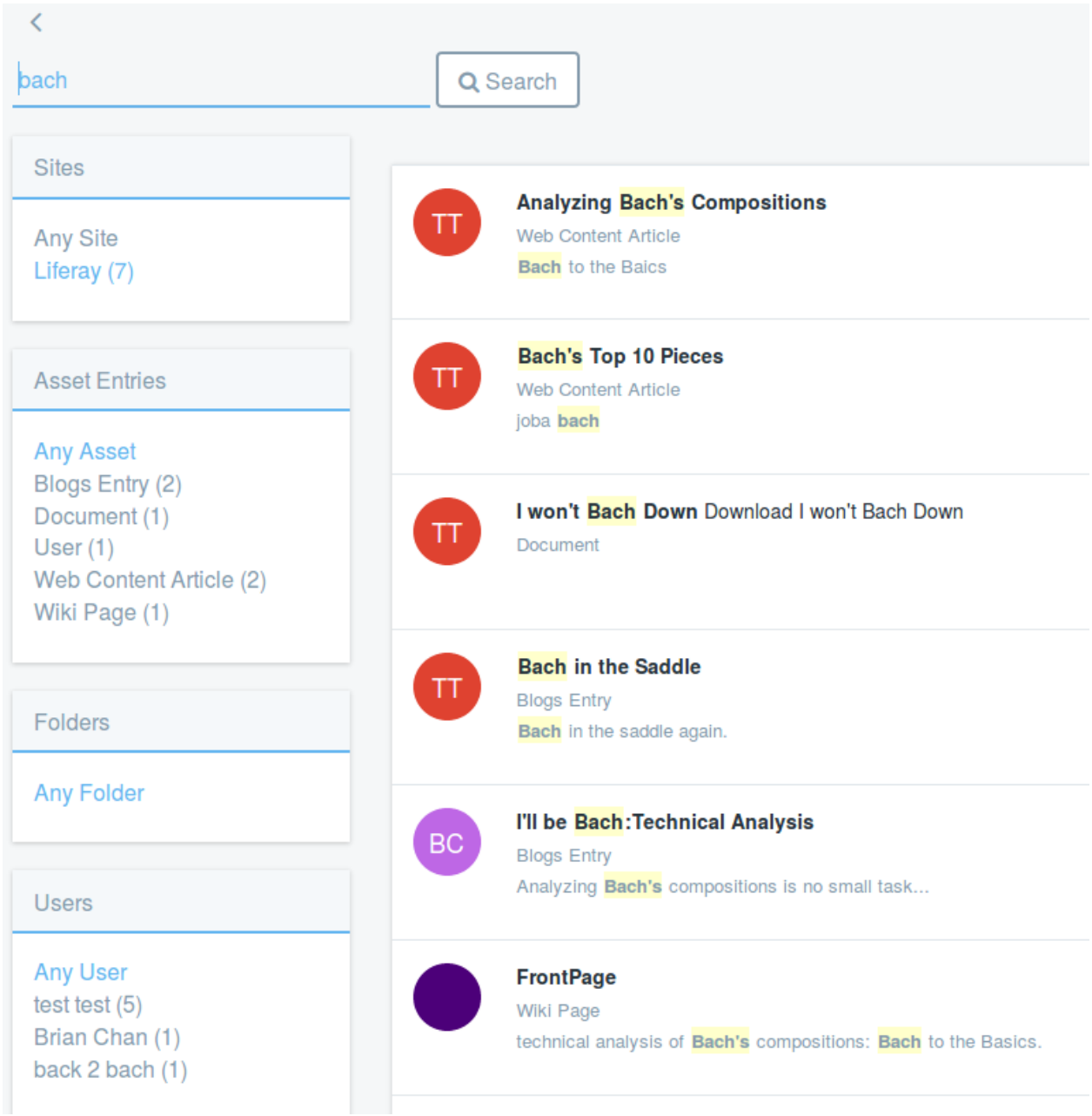


Figure 21.20: When presented with lots of search results, facets are used to narrow down the results list so users can find relevant content.

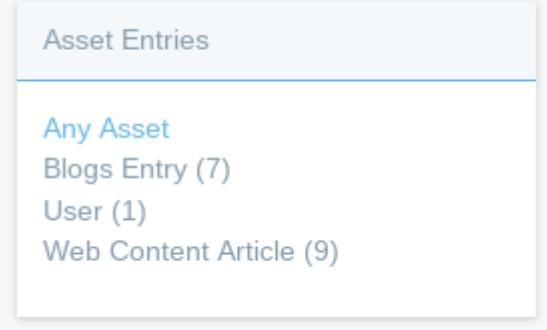


Figure 21.21: Click the *Any Asset* link to clear the filtering for a facet. Now all the available asset types are visible.

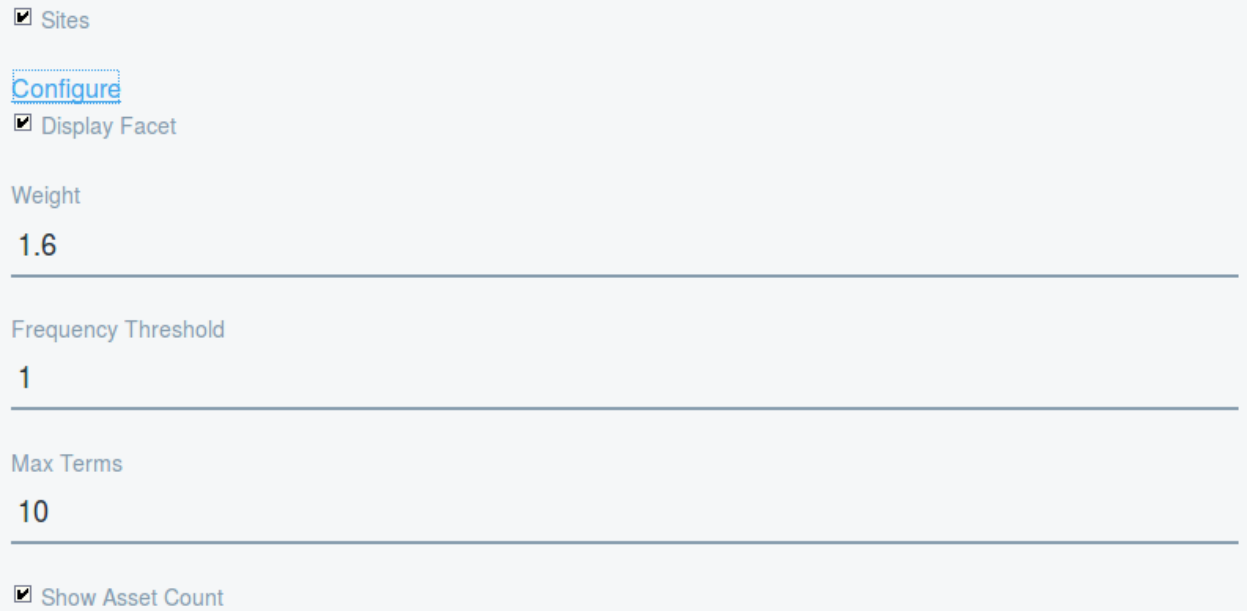


Figure 21.22: Click a facet's *Configure* link to expand its list of settings.

Weight A floating point (or double) value used to determine the order facets appear in the search application. Facets with the largest values are positioned at the top.

Frequency Threshold The minimum frequency required for terms to appear in the result list. For example, if the frequency threshold of a facet is set to 3, a term appearing twice won't appear in the term result list.

Max Terms The maximum number of terms included in the search result regardless of how many matching terms are found for the facet.

Show Asset Count Display the number of terms returned for the facet in the search results.

Current/Available Assets (Asset Type facet only) Add or remove asset types to be included in the Asset Type facet.

Display Style (Tag facet only) Display tags as a Cloud or List (default).

Label (Modified Date facet only) The language key used for localizing the title of the facet when it's rendered.

Range (Modified Date facet only) Define an interval within all the matching terms frequencies' are summed.

Asset Tags and Categories

If tags or categories have been applied to any asset that appears in the result set, it may be displayed in the Asset Tag or Asset Category facet, respectively. Similarly to asset types, not all tags necessarily appear. There may be many more than the 10 tags listed, but the default configuration for this facet is to show the top 10 most frequent terms. As with asset types, this can be modified by setting the `Max Terms` property, described above.

There's no use in having great content if your site's users can't find it. The search functionality included in Liferay DXP is robust, configurable, and easy to use. It accounts for cases that include an overwhelming number of results, very few results, and anything in between.

Chapter 22

Forms and Lists

When you need a form, what you're really looking for is data. Liferay DXP has two applications for building forms to collect precisely the data you need:

1. Liferay Forms: Liferay DXP's primary form building application is for the simplest one or two question survey to the most complex, multi-page, homeowners insurance application containing rules and lists populated by a REST data provider.

2. Dynamic Data Lists (DDL): Provides a user interface tool for building reusable form- and list-based applications intended for display on pages, using templates.

Kaleo Forms: If you're a Liferay Digital Enterprise customer, there's a third form building tool called Kaleo Forms. It integrates form building with workflow to create form-based business processes, like a Conference Room Checkout Form, or a Support Ticket Process so support tickets go through the proper channels on their way to resolution. Read more about Kaleo Forms in the workflow section.

22.1 Which Form Builder Should I Use?

Liferay Forms (also referred to as Forms) is a relatively new application, first appearing in Liferay DXP version 7.0. It's the default form builder in Liferay DXP. If you can use Liferay Forms for your use case, you should.

So the question "Which form builder should I use?" can be restated to "When should I use Dynamic Data Lists?"

- Use Dynamic Data Lists (DDL) if you need a way for users to enter data, *and* you need to display the data in the user interface.

- Use DDL if you need to style your lists and forms with templates.

- Use DDL if there's a field type you need that's not included (yet) in Liferay Forms. These are the field types included in DDL that *are not* in Liferay Forms at the time of this writing: - Geolocation - Web Content - Documents and Media - Link to Page

It's important to note that these (and more!) form field types will be included in future versions of the Liferay Forms application.

Liferay Forms gives you a lot of form design flexibility, whether you need a simple form or a complex form with advanced features. Form pages and form field layout tools make form layouts flexible. While these elements are enough for most forms, more features are planned for Liferay Forms, and the key features of DDL will be merged into Liferay Forms as well. Once that happens, you'll have a single form building application, Liferay Forms. If you're a heavy DDL user, don't worry. Liferay is committed to preserving the core DDL features by integrating them into Liferay Forms.

When all form building features are fully merged into Liferay Forms, the best features of DDL, all the new features of Liferay Forms, and all future improvements will be in one application. Now is the time to familiarize yourself with Liferay Forms and begin using it for all your form building needs, except for the narrow use cases described above.

Chapter 23

Collecting Information from Users

Collecting information from your site's Users is important, whether you're asking them to submit a brief survey or apply for a mortgage. In either case, you need to design a form. Liferay gives you almost limitless form building capability with the Forms application (also referred to as Liferay Forms). For a complete list of form fields available in Liferay DXP, visit the form fields reference article.

Because the complexity of use cases for the Forms application varies from needing only one basic input field, to needing many pages of fields with many different configurations, it makes sense to show you how to quickly build simple forms that can be ready for use very quickly, and show you a case where a form is long, complex, and needs to be well planned in order to capture the needed information. Here's a sampling of the what the Forms application can do right now (yeah, there's even more planned for future releases):

- Populate a Select or Radio field with a REST Data Provider
- Make a field appear based on the value entered into another field
- Add extra pages to the form
- Enable CAPTCHA for a form
- Store results in JSON
- Enable workflow for the form
- Redirect to a different URL after a successful form submission
- Send an email notification to administrators whenever a form is submitted
- Provide a default value (entered if left alone by the user) or a placeholder value (not entered if left alone by user) for each field
- Validate fields using a number of different criteria

If you only need a very basic form, don't worry. Despite this long list of more complex options, developing a simple, elegant form to suit basic needs takes little effort. The next article will cover basic form building, and will be followed by a more complete, complex tutorial on the Forms application.

23.1 Basic Forms

It's really easy to create simple forms with the Forms Application.

At The Lunar Resort, it's important to capture guest's feelings about their stay at the resort. After a (hopefully) safe journey home, guests should receive an email with a link to a page in The Lunar Resort site

with a brief survey that prompts them to rate their stay from a list of selections, and add any additional comments they'd like in an optional field.

You can develop a form like this before lunch time, even if that's in 5 minutes.

Building a Form

Whether you're creating a simple or complex form, it all starts in the same place: the Forms Application in your site's *Content* section. You access this in the Menu by using the site selector to choose the site you want to work in (The Lunar Resort in this case) and clicking *Content* → *Forms*.

- Once you're in the Forms App, click the Add button ( ). You're taken to a form that you'll fill out to create a form. If you're thinking, "Wait, did they just tell me to fill out a form to create a form?" Yes, we did.
- The first step is to give the form a name. For this example, enter *Guest Survey*.
- For Help Text enter *Tell us how your stay was!*
- We need two simple fields, so click the (large) button at the bottom aptly labeled *Add Field*.

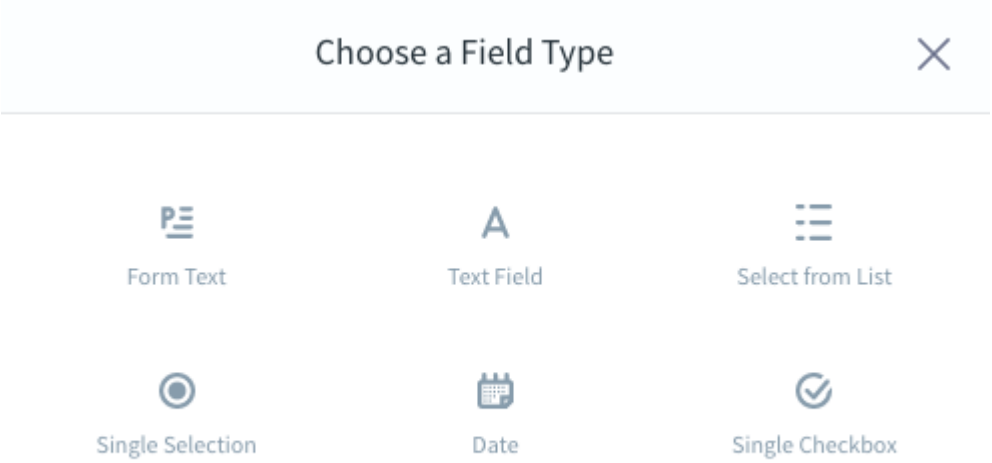


Figure 23.1: You can choose from six field types when creating forms.

- Choose a *Select* field.
- Here's how to configure your select field:
  - **Label:** *Rate your visit to The Lunar Resort.*
  - **Help Text:** Leave this blank for our purposes. If you want a subheading for your field to provide additional guidance, this would be useful.
  - Make sure you turn on the *Required Field* selector. You want to make sure that, at a minimum, you capture whether the guests had a favorable stay or not.
  - Leave the manual option checked for creating the list of selections. If you're interested in populating the field from a data provider, read the next article in this section.

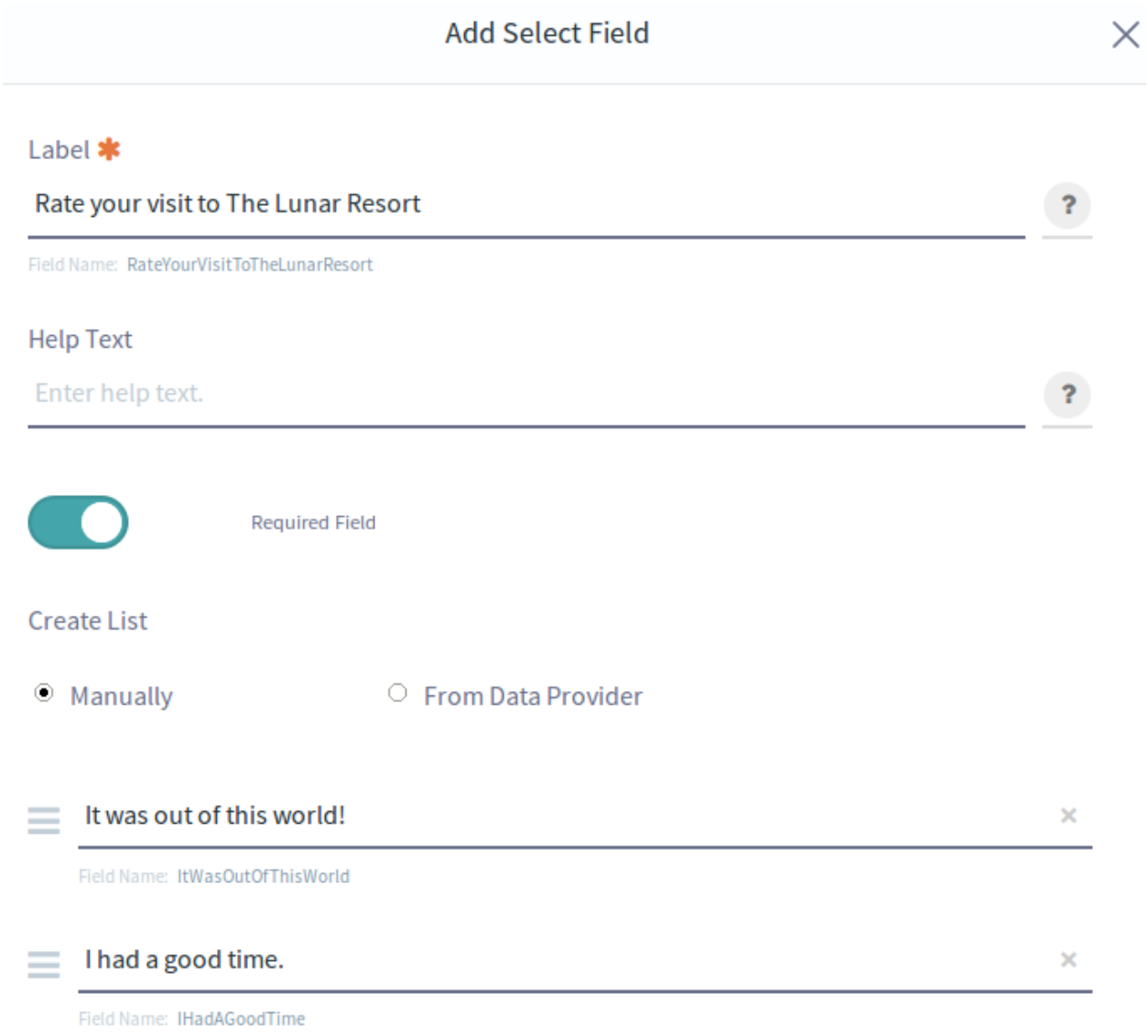


Figure 23.2: Fill out a select field with as many options as you need.

  - Add these options: *It was out of this world!*, *I had a good time.*, *I'd rather go to the beach*, and *I'll never come back*. You'll notice that typing in one of the fields automatically adds another blank selection line. Just leave the last one blank when you're done.

- When you're finished configuring the field's basic options, click *Save*. If you want to see additional options, click *Show More Options*.
- Now you need a text field, so add one using the same procedure you used for the select field.
- Configure the field as follows:

  - **Label:** *Want to tell us more?*
  - **Help Text:** Leave this blank again so the form has a consistent look.
  - **My text field has:** Choose *Multiple Lines*. Let guests prattle on about their stay if they want to.
  - **Required Field:** Leave this unselected. Only require guests to fill out the select field and leave this one as optional.

- Click *Save* when you're finished configuring the text field.
- Back on the form builder page, you can see the way the fields will be laid out. This view shows the first (and only in this case) page of the form.
- This form is finished, so click *Save*.

Now your form can be added to a page, and Lunar Resort guests can be emailed and provided with a link to the page where the form is displayed.

Displaying Forms

The form is developed. Now you just need to display it in The Lunar Resort site.

- Create a new page in The Lunar Resort site called *Guest Survey Page*. Hide it from the navigation, and give it a *1 Column* layout.
- Navigate to the Guest Survey Page and add a Form application to the page.
- Click *Select Form*. When the configuration window opens, click your *Guest Survey* form and then *Save*.
- Your form is now added to the page.

Next learn how to view the form entries. Since there aren't any yet, fill out and submit the form a few times.

Viewing Form Entries

When users fill out your forms, they're generating data. If you're collecting data, you're going to want to see it at some point.

- From the Menu, navigate back to the *Content → Forms* section of The Lunar Resort site.
- Click the *Actions* (⋮) button and select *View Entries*.

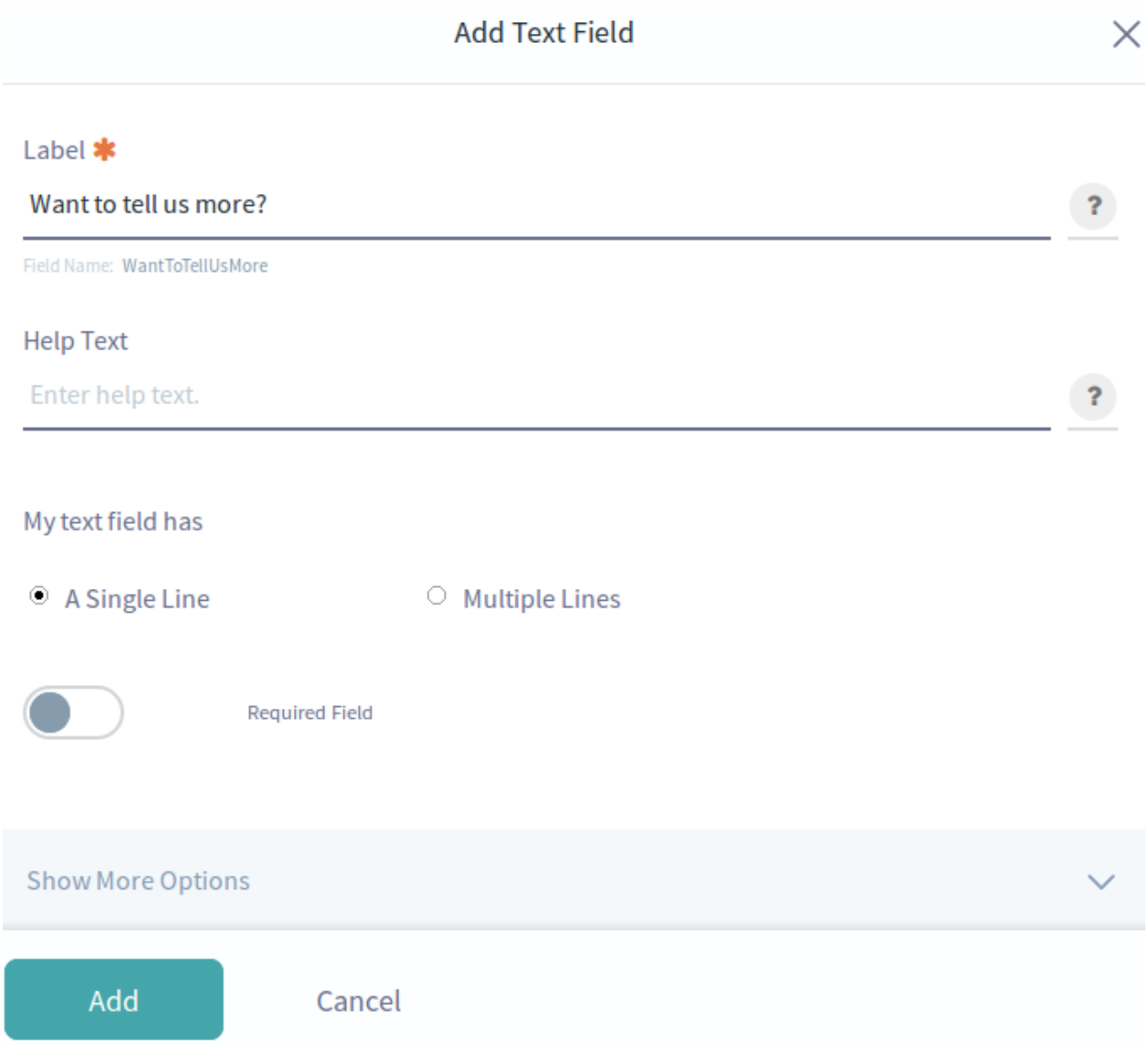


Figure 23.3: Add a text field for guests to provide a longer comment.

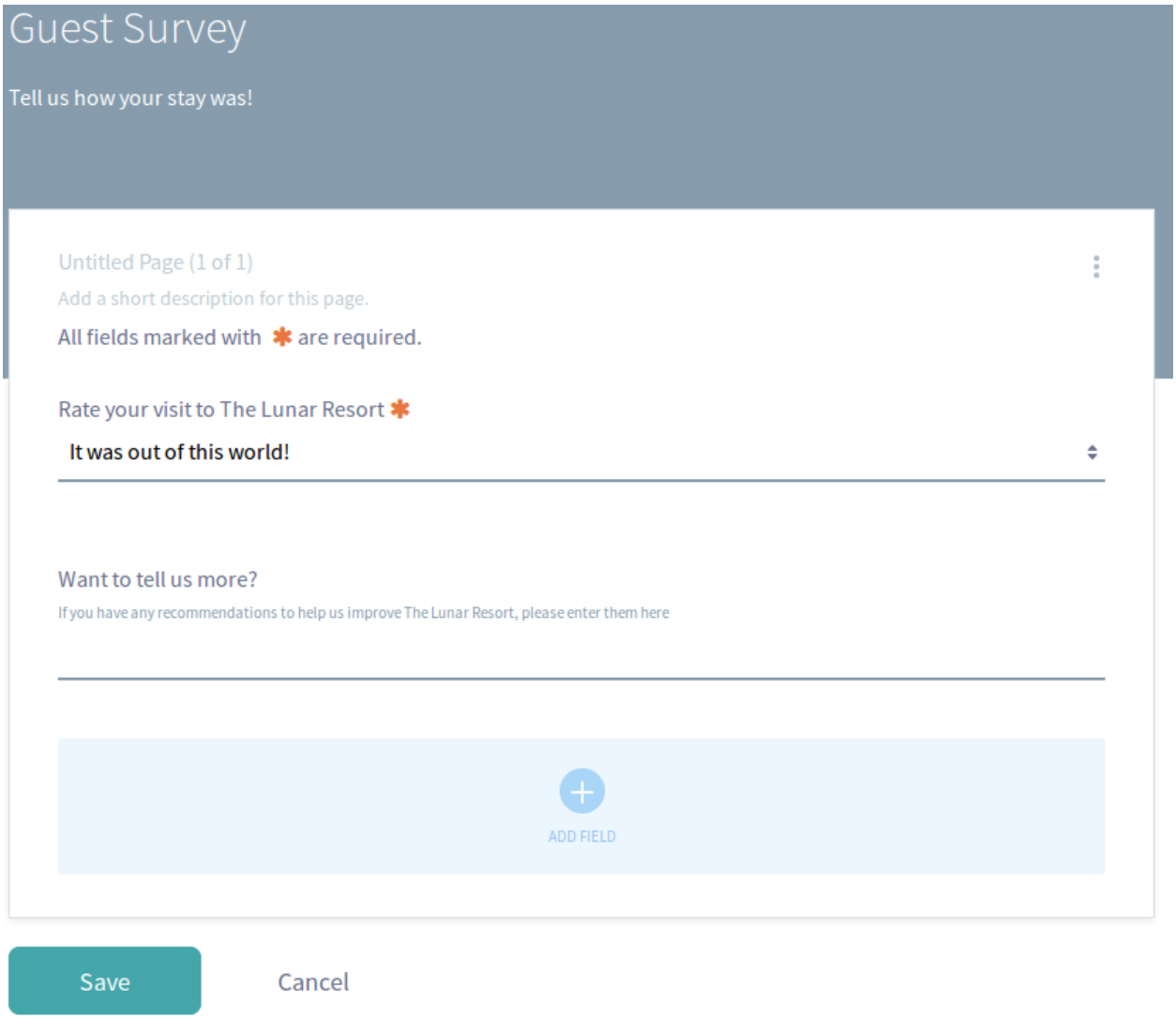


Figure 23.4: The form builder page lets you preview your form layout, add a page to the form, or add some more fields.

- What if you have a lot of form fields and you can't see all of the data for each entry in the search container? Just click the *Actions* (⋮) button for the entry and select *View*. You'll be able to see all the specifics for that form entry.

Viewing entries is great, but this is serious data we're talking about. You might need to get all the entries into a spreadsheet so you can work with them, for example.

Exporting Form Entries

So, you need to put your form entries in a spreadsheet to do spreadhseet things with them? No problem.

- Navigate to the Forms application in The Lunar Resort site's Content section again.
- Click the *Actions* (⋮) button and select *Export*.

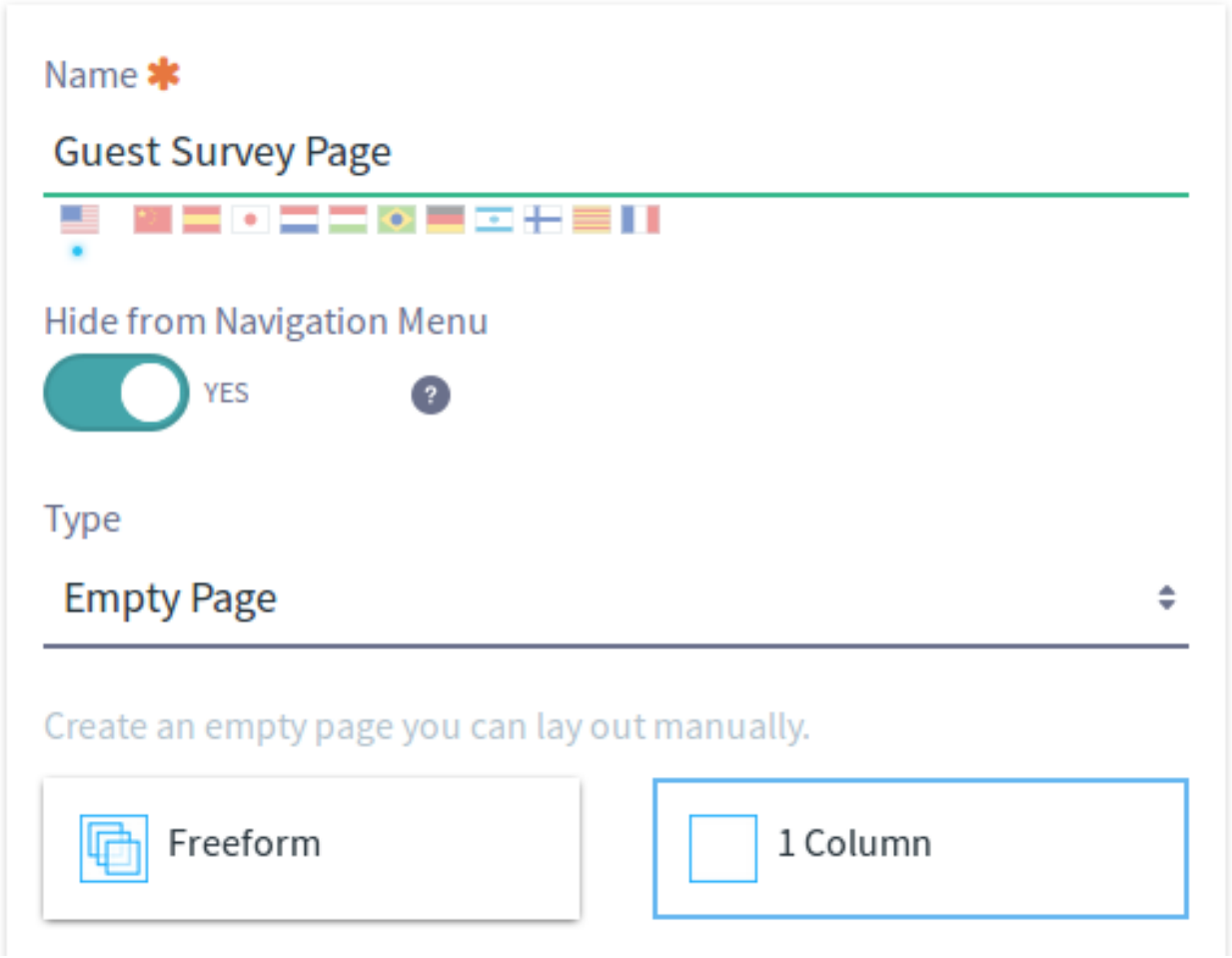


Figure 23.5: Add a page for guests to view and fill out your new form.

- Choose a File Extension. You can get entries in CSV or XML. For this example, pick CSV.
- Click *Okay*, and open the file or save it locally. Open it with your favorite spreadsheet program and confirm that your form entries are entered into a spreadsheet for you.

Note: The Forms application itself has an *Import/Export* window that you can access from the application's Configuration menu ( ). This is how you import and export the application configuration and its data (forms and form entries). The file format for this type of import and export is a LAR file. For more information, see the article on importing and exporting application content..

Once you export a batch of form entries, it can make sense to delete them from the database.

Deleting Form Entries

What if you export a form's entries and now you want them to be removed from the Liferay database? It's easy to delete all of a form's entries at once.

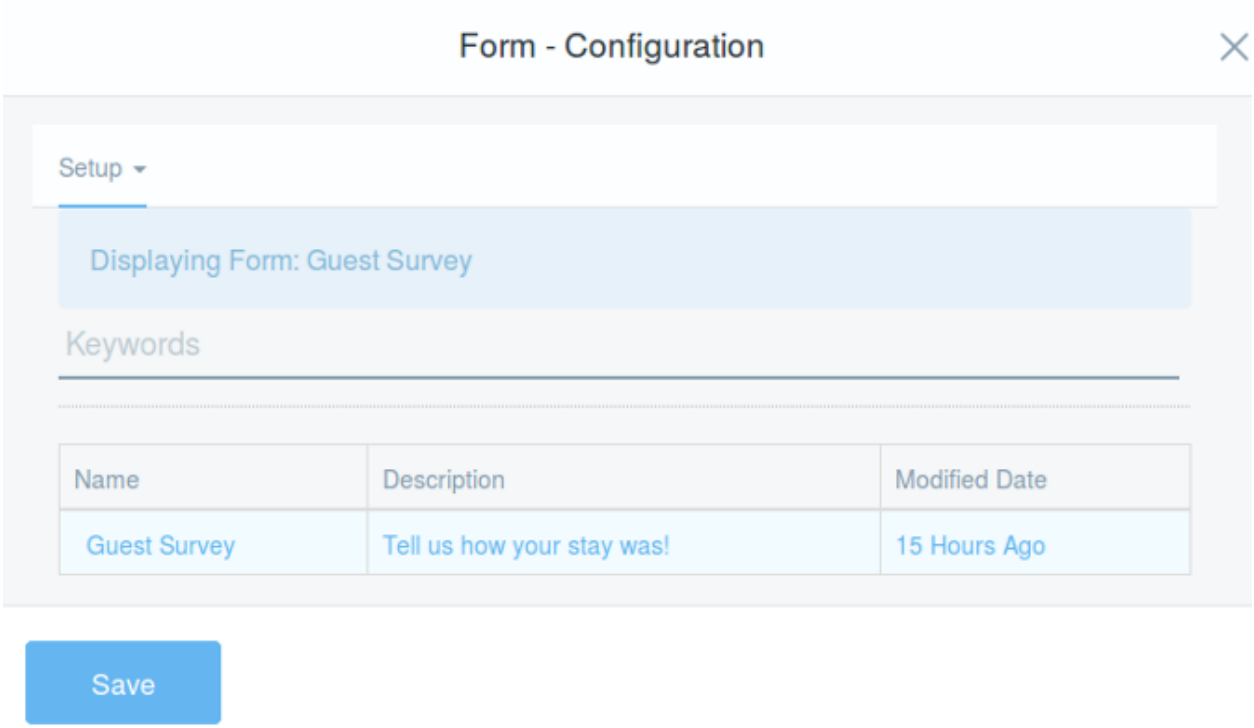


Figure 23.6: Select the form you want to display.

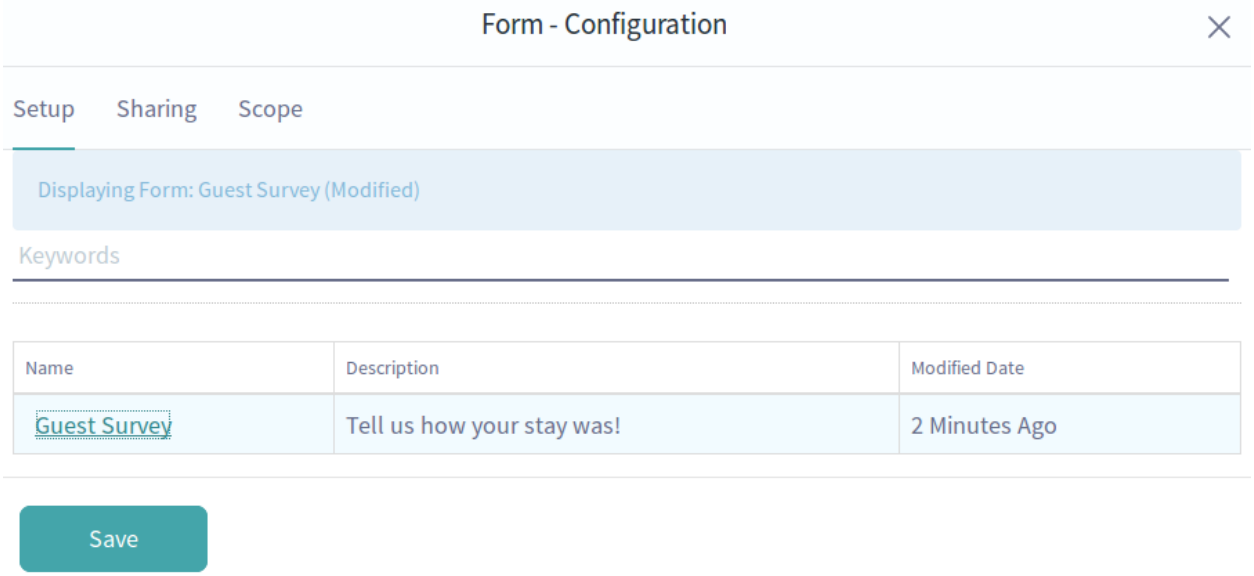


Figure 23.7: Lunar Resort Guests can access a simple form to record their feelings about the resort.

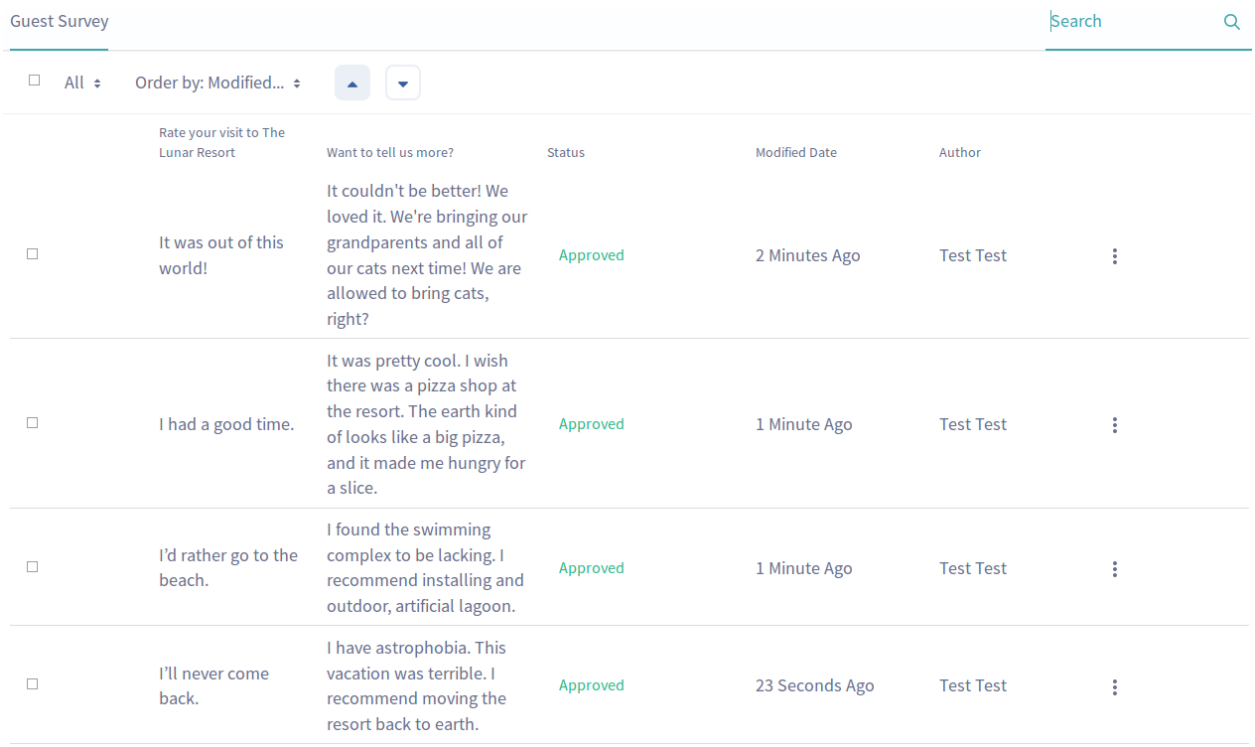


Figure 23.8: You can view the entries right in the Forms application.

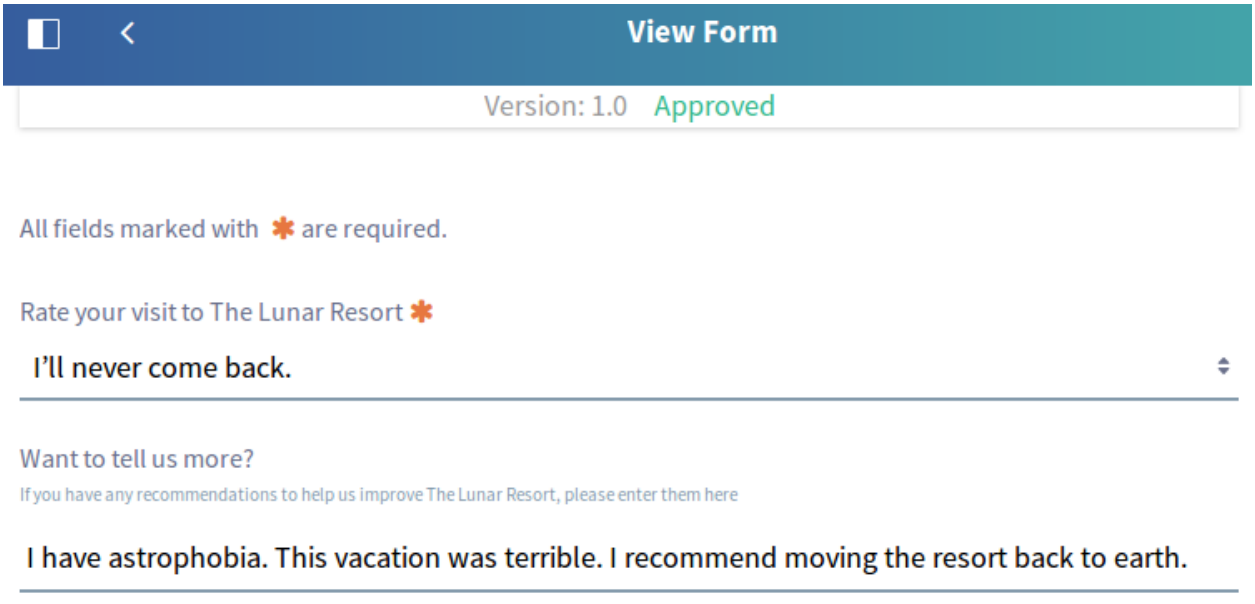


Figure 23.9: You can view a single entry right in the Forms application.

	A	B	C
1	Rate your visit to The Lunar Resort.	Want to tell us more?	Status
2	I'll never go back.	I have astrophobia. This vacation was terrible. I recom	Approved
3	I'd rather go to the beach.	I found the swimming complex to be lacking. I recomm	Approved
4	I had a good time.	It was pretty cool. I wish there was a pizza shop at the	Approved
5	It was out of this world!	It couldn't be better! We loved it. We're bringing our gra	Approved
6			

Figure 23.10: : You can export entries as CSV or XML files.

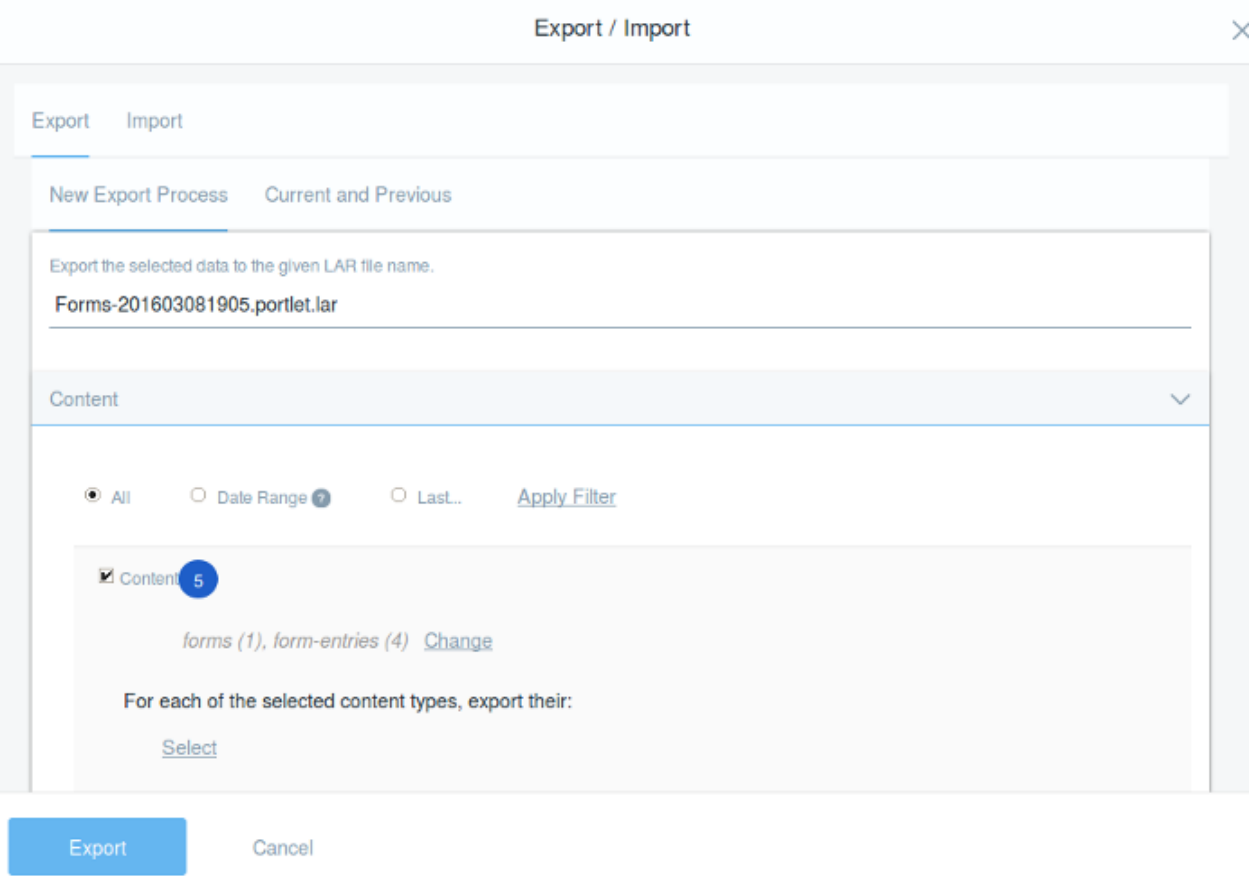


Figure 23.11: : As with other Liferay Apps, you can import or export forms and their entries with a LAR file.

Navigate back to the Forms application In The Lunar Resort Content section. Click the *Actions* (⋮) button next to the Guest Survey form and select *View Entries* again. Select all of the entries by checking the box next to *All*. A trash can icon ( ) appears in the top right corner of the Form Entries screen. Click it.

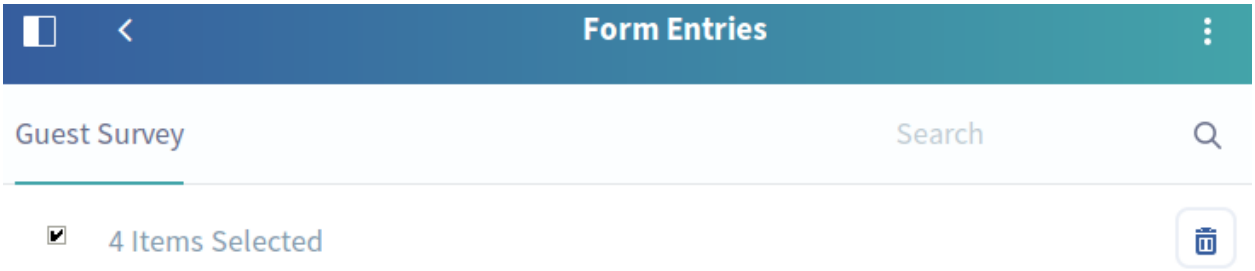


Figure 23.12: : Delete all form entries in one fell swoop.

If you just wanted to delete a single entry, you'd just select that entry by checking its box, then click the trash can.

If you're worried about deleting everything irrecoverably by accident, don't worry. You'll have to confirm the deletion in a dialog box that pops up after clicking the trash can.

Now you know the basics of creating and managing forms in Liferay. But this presentation didn't do the Forms application justice. It's much more powerful than hinted at here. Read the next article to see what the Forms app can do for your more advanced form buiding needs.

23.2 Creating Advanced Forms

Liferay's built-in Forms application supports your most basic form building needs (something like a one question survey), and your most advanced (like a mortgage application, or conference sign-up form). In a separate article, learn about the basics of form creation:

- Navigating to the Forms application
- Adding form fields, and supported field types
- Configuring form fields
- Viewing and Exporting form entries
- Deleting form entries

This article covers more advanced use cases for forms, and a basic knowledge of the Forms application is assumed. Here's a current list of the features you can use in your forms that are covered in this article:

- Populating a Select field with a REST Data Provider
- Making a field appear based on the value entered into another field
- Add extra pages to the form
- Providing a predefined value or a placeholder value
- Validating fields using a number of different criteria
- Configuring form permissions
- Laying out the form

Aside from those features, which enhance the building of your forms, there are some settings and processing features to explore:

- Enabling CAPTCHA for a form
- Redirecting to a different URL after a successful form submission
- Sending an email notification whenever a form entry is submitted

---

Note: By default, form entries are stored in Liferay's database using JSON format. If you navigate to Form Settings (click the *Options* ( ) button → *Settings*), you'll see that the only storage option is JSON. Keep in mind that developers can implement their own storage types if JSON is not desirable. If they do, the option will appear here in Form Settings.

The Lunar Resort Employment Application Form

At The Lunar Resort, management is VERY particular in its hiring practices. Lots of people think they want to work on the moon full time, but it takes a special kind of person to really make it work. There's so much that can go wrong:

- Astrophobia: If a newly hired employee discovers they have an unshakable fear of space, it's quite costly to send them back home.
- Claustrophobia: The tiny sleeping capsules where employees rest is not compatible with a fear of small spaces.
- Alienophobia: Some of The Lunar Resort guests are not from planet Earth. It's important for employees to be comfortable and friendly with the more "exotic" guests, such as the Daleks.

There are more hiring pitfalls than the phobias listed above, but it's time to start building the application form, starting with the creation of its pages.

Creating Form Pages

Liferay's Forms application supports multi-page forms. To add one, click the Actions ( ) button at the top right hand corner of the form, and then click *Add New Page*.

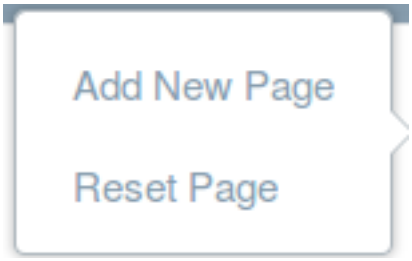


Figure 23.13: You can add new pages or reset the current page from the Page Actions menu.

You also have the option to *Reset Page*. If you click this, the title, description, and any fields you added to the page will be deleted. It gives you a blank slate.

The Lunar Resort employment form needs three pages with clear page descriptions:

- Personal Details: This page requires you to give us some pretty personal information.
- Very Personal Details: This page requires you to give us some very personal information.
- Extremely Personal Details: This page requires you to give us some extremely personal information.

Once you create the form's pages, you'll see how navigating the pages is easy with the form page navigation wizard.

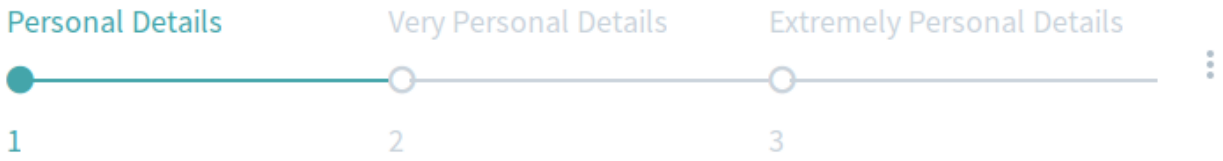


Figure 23.14: Multi-page forms have a default *Wizard Mode* that makes navigating the form easy for users.

Once you have the form pages created, navigate back to the first page, Personal Details.

Adding Placeholder Text and Predefined Values

To populate the employment application form for The Lunar Resort, you should start by adding some of the usual fields, like a Full Name field:

- **Full Name:** Add a required text field titled *Full Name*, with a description that reads *Enter your full name, no matter how long*. Under Show More Options, enter Placeholder Text: *e.g., Maximillian Aurelius Piroux the 11th*.

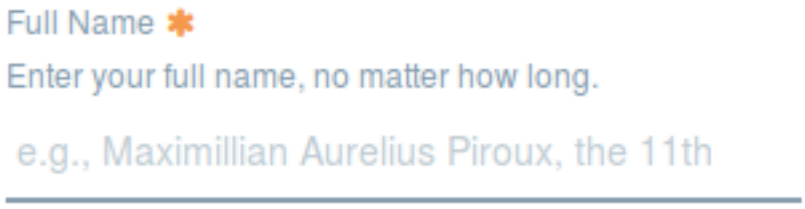


Figure 23.15: The Full Name field in the employment form. It has a placeholder value to help users understand what you are looking for.

Note: There's an important difference between *Predefined Values* and *Placeholder Text*. If a field is left untouched by the user, the predefined value will be submitted, while placeholder text is not submitted (it's as if the user left the field blank).

Remember that placeholder values will not be submitted if the field is left blank, so you don't have to worry about getting a bunch of applications from *Maximillian Aurelius Piroux the 11th*.

Validating Text Fields

The HR department needs to weed prospective employees out if they're not mature enough. If the user is under a certain age, make sure to ask them if they've ever been away for an extended period of time. This can be done using a text field that accepts only numbers (age), and then a field to probe whether the applicant can handle being away from home for a long period of time. In that field, you can use a Field Visibility Expression so it only appears if the user enters a number less than 30.

- **Age:** Add a required text field called *Age*, with a description that says, *Enter your age in years*, and the placeholder text *Numbers Only*.

Enable Validation and select *Number*, *is greater than*, and enter 0. You can enter an error message *You must enter your age in years, and it must be greater than 0*.

Figure 23.16: The Age field in the employment form. It uses validation to ensure that a number greater than zero is entered.

Note: Field validation lets you decide if you're going to reject certain types of information from being submitted. If a form field's data doesn't pass the validation rules you specify, the entire form entry won't be submitted until the user enters valid data. What validation rules are built in for your convenience?

- Text: If you begin by selecting *Text*, only text submissions are allowed for your form field.
  - Contains: Selecting *Contains* means you're going to specify a character or word that *must* be entered.
  - Does not contain: If you use this rule, you're going to specify a character or word that *must not* be submitted.
  - URL: Submissions must be URLs with the proper syntax.
  - Email: Submissions must be email addresses with the proper syntax.

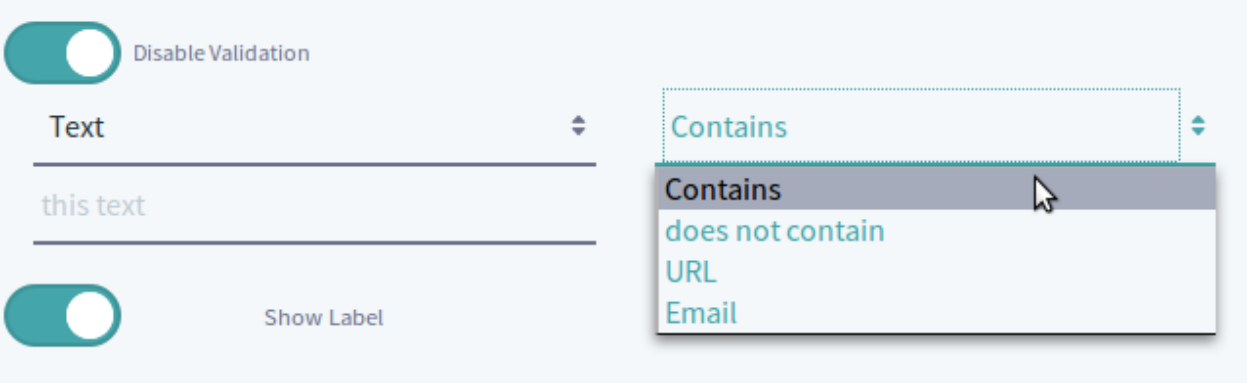


Figure 23.17: You can validate text submissions for text fields.

- Number: If you select *Number*, only number submissions are allowed for your form field.
  - Is greater than or equal to: Specify the smallest number that can be entered.
  - Is greater than: Entries must be greater than the specified number.
  - Is equal to: Allow only entries equal to the number you specify.
  - Is less than or equal to: Specify the largest number that can be submitted.

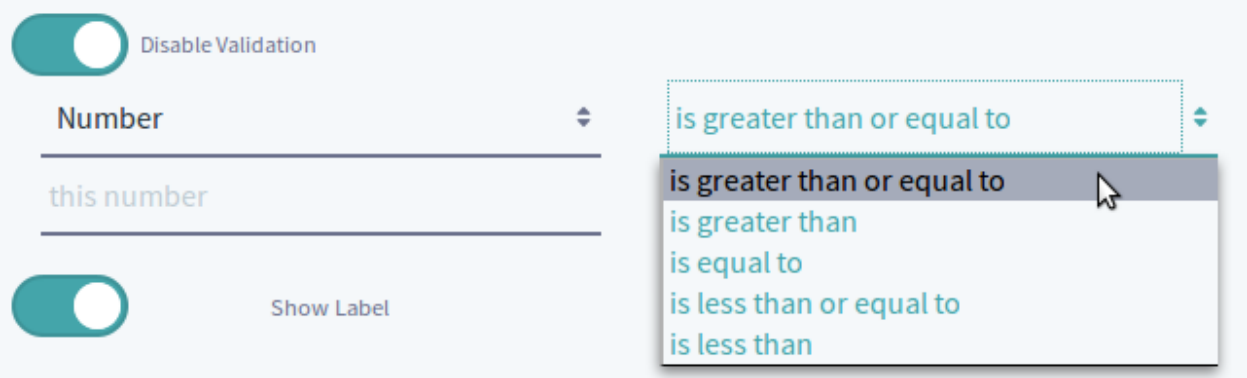


Figure 23.18: You can validate number submissions for text fields.

– Is less than: Entries must be less than the specified number.

It can be helpful to alert users to your validation rules in help text or placeholder text (for example, *You must enter a number greater than 0*).

If you mark a field as *Required*, that's another form of validation. Liferay is going to make sure that the field is not empty. If it is, an error message will appear, directing the user to enter something into the required field.

Make one more required field on the first page:

- **Email Address:** A required text field that only accepts email address. Use the validation rule for *Text → Email* to make sure that a valid email address is entered.

Next create a field that will be displayed only if the user enters any number less than 30 in the Age field.

Using Field Visibility Expressions

Sometimes you only want a form field to appear if a certain value is entered in another field. In the Lunar Resort application for employment, you want to ask an additional questions to users who are less than 30 years old, to make sure they're comfortable with being away from home for a long period of time.

- **Maximum Time Away From Home:** This is a required select field with the description *What's the longest time, in consecutive months, that you've spent away from your permanent residence?* Give it the following options: *<1*, *1-3*, *4-12*, *>12*.

  Enter a Predefined Value of *<1*.

  Now comes the interesting part. Under Field Visibility Expression, enter *between(Age,0,29)*. This expression will cause the field to appear *only* if the Age field has a value between 0 and 29 entered. Another expression that will produce the same result is *(Age>=0)&&(Age=<29)*. In this case, the expression is checking to make sure that the Age value is greater than or equal to zero and it's checking to make sure that the entered Age value is less than or equal to

29.

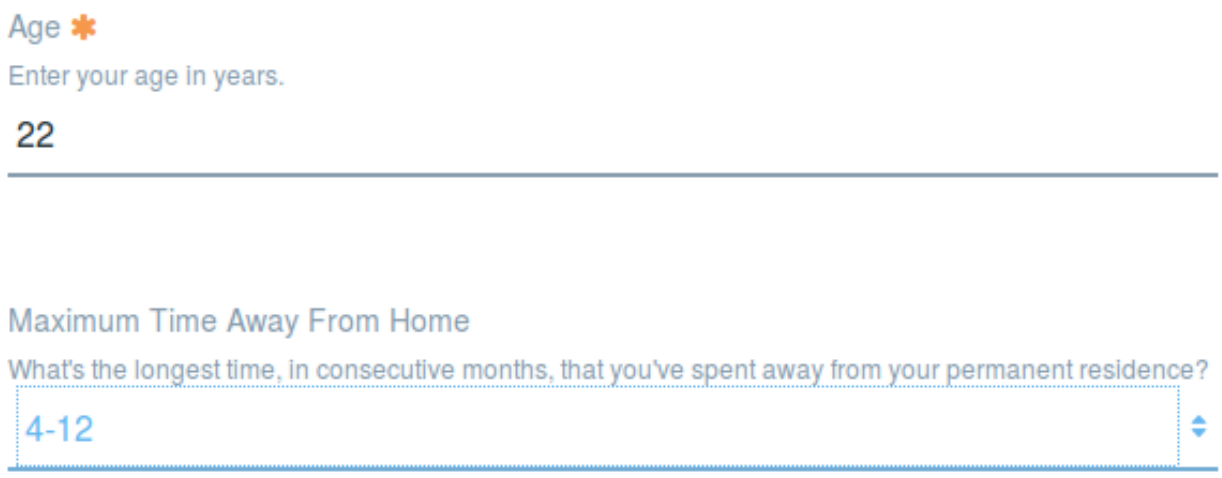


Figure 23.19: You can make a field appear based on the value of another field in your form.

Note: There are several functions and operators you can use to control field visibility based on the value entered in another field. An expression can be as simple as containing a single operator:

```
Age<29
```

Alternatively, operators and functions can be combined to make a more complex expression:

```
sum(Housing,CarPayment)>MonthlySalary
```

In this expression, the field would only be displayed if the sum of the *Housing* field and the *Car Payment* field is greater than the *Monthly Salary* field.

As demonstrated above, you can use a form field value (whether text or numeric) in place of a hard coded value, by entering the Field Name instead of a specific value.

Figure 23.20: A form field's Field Name is automatically generated from the Label you give the field.

There are functions and operators for your use in your field visibility expressions. First consider the functions:

- **between(FieldName,lower bound value, upper bound value)**: Check whether the entered value of the specified numeric field is between a lower and upper bound. If it is in the specified range, the field will appear on the form. For example:

```
between(Age,0,29)
```

- **equals(FieldName, value)**: Display your field only if the value entered into the specified field equals a specific numeric value (or if it equals the value of another field). For example:

```
equals(Salary,TotalExpenses)
```

- **sum(FieldName1, FieldName2, ...)**: Add numeric fields together, usually for comparing to a given value using one of the operators covered below. For example:

```
sum(Expense1,Expense2,Expense3)>Salary
```

- **min(value, FieldName1, FieldName2,...)**: Get the lowest numeric value in the list of parameters (you can use Field Names and/or values), usually for comparing to a given value using one of the operators below. For example:

```
Salary>min(TotalExpenses,TotalDebt)
```

- **max(Field Name, value)**: Get the largest numeric value in the list of parameters, usually for comparing to a given value using one of the operators covered below. For example:

```
Salary<max(TotalExpenses,TotalDebt)
```

- **contains(Field Name, value)**: Used to check a field for a specific string of text or the text entered in another field. The text value can be in the beginning, middle, or end of the field being checked. For example:

```
contains(FullName,LastName)
```

- **concat(Field Name, value)**: Concatenate a string of text with a given text value, usually used in constructing a larger expression. For example:

```
contains(FullName,concat(FirstName, " Smith"))
```

In addition to the functions described above, there are a bunch of operators you can use in building your Field Visibility Expressions:

- **+**: Addition operator
- **-**: Subtraction operator
- *: Multiplication operator
- **/**: Division operator
- **%**: Division remainder operator
- **^**: Power operator
- **&&**: *And* condition operator
- **||**: *Or* condition operator

- **>**: *Greater than* relational operator
- **>=**: *Greater than or equal to* relational operator
- **<**: *Less than* relational operator
- **<=**: *Less than or equal to* relational operator
- **==**: *Equals* relational operator
- **!=**: Relational operator to make sure the arguments are *not* equal
- **NOT**: Negates a boolean expression. Boolean (true or false) expressions check for a *true* condition by default. You can use *NOT* to check the opposite:

  ```
  NOT(contains(FullName, FirstName))
  ```

  The above expression checks whether the *FullName* field contains the value entered in the *FirstName* field. If it does *not* then your field will appear.

To check for a value in a select field, you need to use the Field Name for the field, and as the value, use the Field Name for the select option you entered.

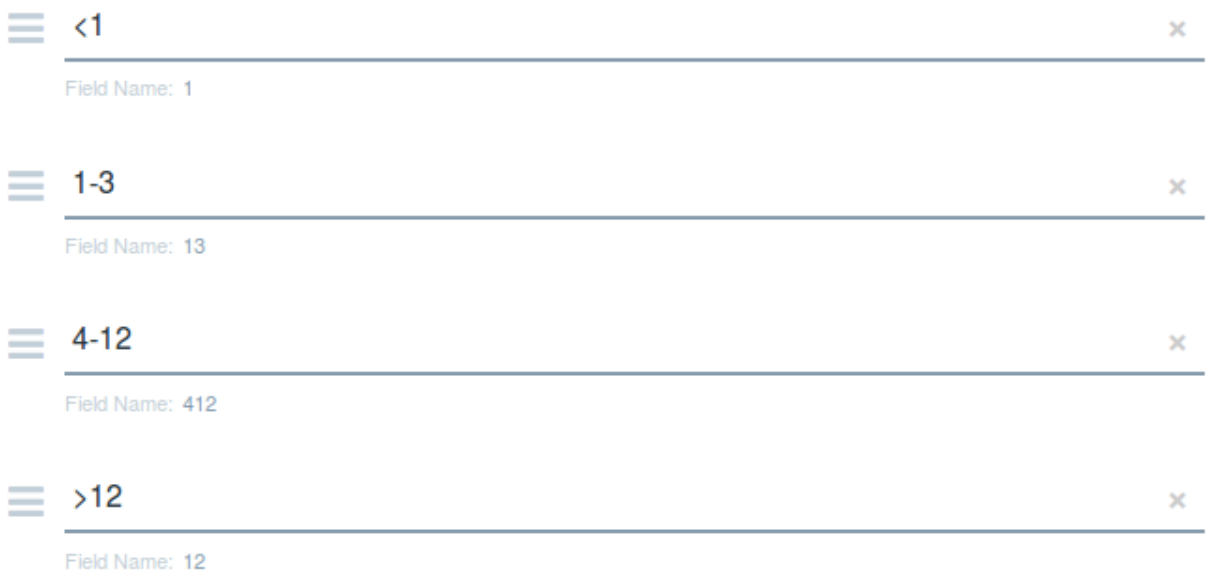


Figure 23.21: A select field's options will be given Field Names. These are used as the value in your Field Visibility Expression.

For example, if you want to display a field only if the *Maximum Time Away From Home* field's *>12* option is selected, you could write this Field Visibility Expression:

```
contains(MaximumTimeAwayFromHome,12)
```

Field Visibility Expressions are great for dynamically displaying a field based on another field's entered data. Next, find out how to dynamically populate a select list from a Data Provider.

Populating Select Fields with REST Data Providers

On its final page (*Extremely Personal Details*) The Lunar Resort application form needs five select fields, all populated with the complete list of the countries of earth. Read the sidebar note below to find out why this field is needed (it's just a fictional story, so skip if you don't have time for some light entertainment).

Note: Interstellar citizenship rules are strange, and since the moon declared independence from earth (Lunar Independence Day is a big deal at The Lunar Resort), it has its own set of citizenship rules. Among other things, the rules require employees of The Lunar Resort to be citizens of the moon.

Moon citizenship is cool, especially if you're a dual citizen of your native earth country and the moon! However, during the Space Wars (in which the moon gained its independence), many countries on earth made strict laws dictating that dual citizenship would never be granted if the applicant's second citizenship allegiance was to the moon. That can lead to interesting situations when an employee leaves The Lunar Resort, or is terminated. The Lunar Resort management needs to know a leaving employee's top five destinations on earth. If none of the countries the employee lists are willing to accept them, there's a secret colony of earth's expatriates on the dark side of the moon that usually has vacancies.

That's why the application form's last page needs to have five select fields with all the countries of earth.

Unless you have an intern, you don't want to manually enter all the countries of earth into five different select fields. Instead, you want to populate the fields using a Data Provider.

Adding and Configuring a Data Provider Before using a Data Provider in your select fields, you need to configure one that you can use in your forms. Navigate to the Forms application in *Site Administration → Content*, and click the Options ( ) button.

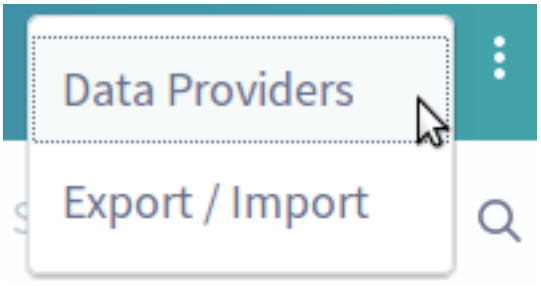


Figure 23.22: : Set up a data provider that can be reused in all your forms.

If you click on *Data Providers*, the data providers window opens, and you can see any existing data providers or add a new one. Click the *Add* button ( ).

Fill out the Data Providers form, passing in the following values:

- Name: *Countries List*
- Description: *A complete list of earth's countries*
- URL: http://localhost:8080/api/jsonws/country/get-countries
- Displayed JSON Attribute: *nameCurrentValue*
- Stored JSON Attribute: *name*
- User Name: test@example.com (or whatever your admin user name is)
- Password: *test* (or whatever your admin password is)

You're probably wondering what the *URL*, *Displayed JSON Attribute*, and *Stored JSON Attribute* fields are all about. First, it's good to understand that what you're doing when setting up a data provider is accessing a REST web service. You can use the JSON web services registered in Liferay, or any other REST web service you have access to. To find a list of the ready-to-use registered JSON web services in Liferay, navigate to http://localhost:8080/api/jsonws (assuming you're running Liferay locally). If you do that, you'll be able to browse the available Liferay services that you can use to populate your list. The services useful to you in the Forms application will get a list of something. The service you are calling in the data provider you set up above gets a list of countries. Find the `get-countries` JSON web service and click on it, then click *Invoke*. The *Result* tab will have a list of countries using JSON syntax, like this:

```
[
  {
    "a2": "AF",
    "a3": "AFG",
    "countryId": "20",
    "idd": "093",
    "mvccVersion": "0",
    "name": "afghanistan",
    "nameCurrentValue": "Afghanistan",
    "number": "4"
  },
    ...
```

That's the record for the country Afghanistan. As you can see in the *URL* tab, the URL you entered into the data provider form is the same as the one generated for accessing the `get-countries` JSON web service. You can find the URL for any registered JSON web service using this same procedure.

The *Displayed JSON Attribute* field is where you specify which field you want to use to populate your select list. In this case, `nameCurrentValue` provides the full, properly capitalized country name, and that's what you want users of the form to see.

The *Stored JSON Attribute* field is where you set which field will be stored in the database for the form entry. The `name` field was selected in this example.

Once you set up the data provider, click *Save*. Now you can set up five select fields that list all the countries of earth in no time.

Using a Data Provider in a Select Field Back in the application form, go to the third page, Extremely Personal Details. Add a select field.

- **Relocation Country: First Choice**: Make the select field required, and enter *If relocation becomes necessary, to which country on earth would you most like to go?* as Help Text. Select the radio button to *Create List From Data Provider*, and choose *Countries List*. Click *Save*.

If you're building the example application form, do the same thing four more times, using field labels to reflect the choice number. For example, you'd call the second select field *Relocation Country: Second Choice*, and so on until you have five of them. You can decide what help text to provide, if any.

Your Data Provider is now being used to populate your select fields. However, this form is going to be submitted by Guest users, who don't currently have permission to see the list of results from the data provider.

Granting Data Provider Permissions To configure the data provider's permissions, navigate to the Forms application (*Site Administration* → *Content* → *Forms*). Open the Options menu ( ) and select *Data Providers*. For the data provider you want to configure, click the Actions button ( ), then *Permissions*.

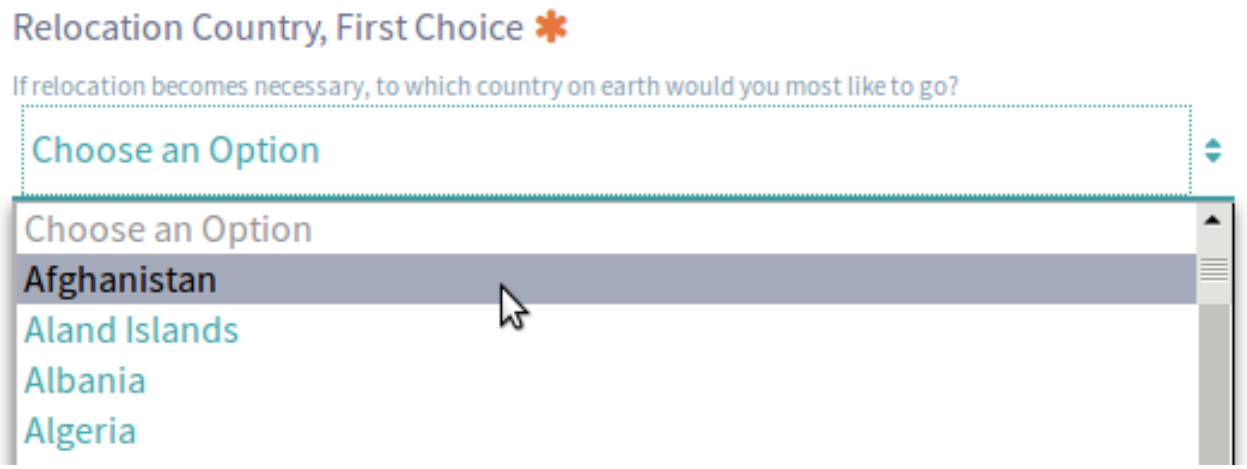


Figure 23.23: : You can configure a data provider and use it to populate a select field.

For the employment application form in The Lunar Resort, Guest users need the *View* permission, or else they won't be able to see the options provided by the data provider in the select fields. Once you grant the permission, click *Save*.

Note: If you're developing The Lunar Resort employment application form, you'll notice we skipped over the second page, Very Personal Details. It will be dealt with in the section on laying out your forms.

That concludes the actual development of the form (adding pages and form fields). The rest of the article covers features of the Forms application that you'll use during or after form entry submission.

Enabling CAPTCHA on Form Submissions

You're probably familiar with CAPTCHA, the program used to prevent a bot from submitting forms. It's often used in Login applications, but you can use it in the Forms application, too.

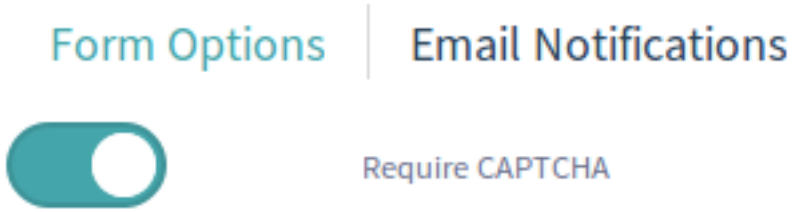


Figure 23.24: : You can enable CAPTCHA for your form in the Form Settings window.

To enable CAPTCHA, click the form's *Options* ( ) button, then *Settings*. Enable the *Require CAPTCHA* setting, click *Done*, save the form, and that's all there is to it!

Now, you're much more likely to receive form submissions from humans!

Figure 23.25: : Once you enable CAPTCHA, your form has protection against bot submissions.

Setting Up Form Notifications

It's possible to have a notification email sent each time a form entry is successfully submitted. Open the form's Form Settings section (click the *Options* ( ) button → *Settings*). Click the *Email Notifications* tab, enable the option to *Send an email notification for each entry*, and fill out the fields:

- From Name: The name of the sender. This could be the site name, the form, name, or anything else that will be informative to the recipient.
- From Address: You can use something like *noreply@example.com*, so that recipients of the notification don't try to reply.
- To Address: This should be the email address of the notification recipient (for example, *test@example.com*).
- Subject: An informative subject line tells the recipient what happened. *An application for employment was submitted in The Lunar Resort* might be appropriate for the form developed in this article.

If you enabled workflow for the form, and there's already a notification going to the people who need to process the form entry, you might not need to generate any notification from the Forms application itself.

Redirecting Users after Form Submission

You can send users to any URL you want upon form submission. You might have a particular *Thanks for your submission!* type page with a link to the home page, or you might want to direct a user somewhere specific based on the form they just submitted.

Whatever your use case is, you can set up a redirect URL by opening the form's Form Settings section (click the *Options* ( ) button → *Settings*). In the *Redirect URL* just type the URL. That's it.

So, your form is all set up ready to be filled out by users. However, you haven't configured the permissions yet, so the only user with permission to add records is you, the owner! There's a good chance you'll want to customize the form's permissions.

Configuring Form Permissions

By default, you can grant the following permissions for a form:

- Delete: Permission to delete the form.
- Permissions: This gives any role with the permission the ability to configure the form's permissions.
- Add Record: Permission to submit form entries.
- Update: Permission to update form entries.
- View: All the other permissions you can configure for the form depend on this one. Think about it: if you can't see the form, how can you add a record?

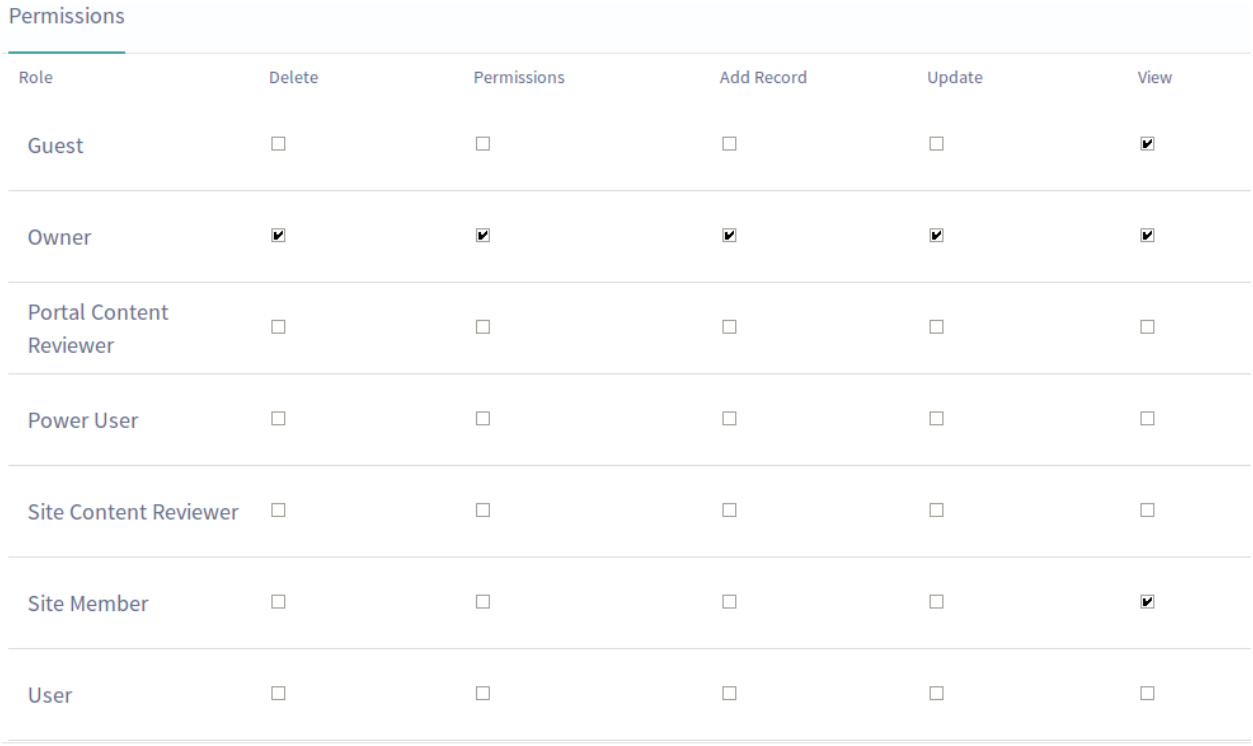


Figure 23.26: : You can configure the permissions for a form.

If you want to allow unauthenticated visitors to your site to view and fill out forms, give the *Guest* role the *View* and *Add Record* permissions.

Navigate to the Forms application (*Site Administration* → *Content* → *Forms*). Click the application employment form's *Actions* button (⋮), and select *Permissions*. Give the Guest role the View and Add permissions, then click *Save*.

Note: By default, all users inherit the Guest role's permissions. The Guest role represents unauthenticated visitors of your site, and it makes sense that if you want to allow Guest users to submit forms, you're fine with your site members and portal users submitting forms, as well. If you want to disable the automatic inheritance of the Guest role permissions, there's a property you can set in your `portal-ext.properties` file:

```
permissions.check.guest.enabled=false
```

Form Layouts

Sometimes, it doesn't make sense to have a single column, vertically oriented form, which is what you get by default when you just keep clicking the large *Add Field* button on the form as you build the form.

Instead, you might use more than one column in your form.

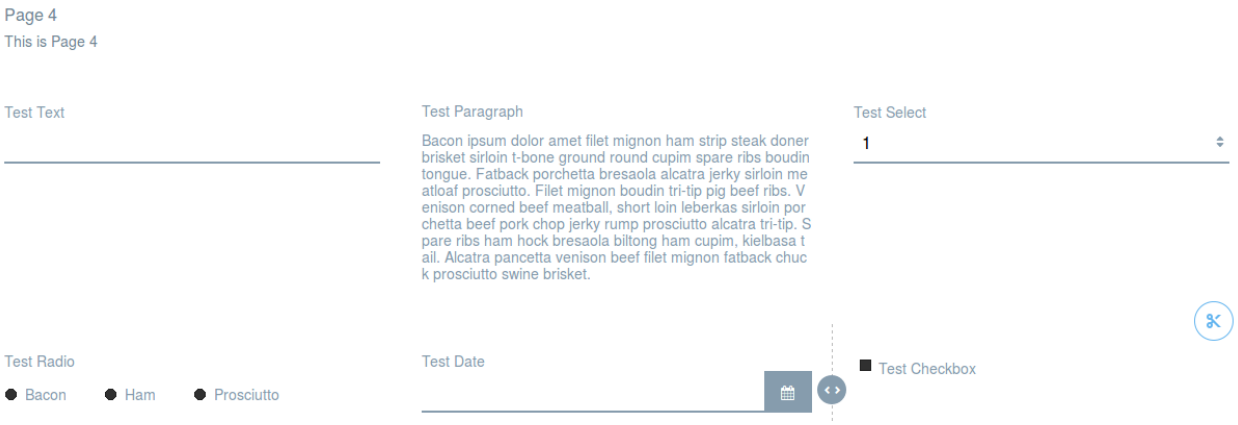


Figure 23.27: : An example of a multiple column form layout.

As you can see, the multi-column approach saves a lot of space in the pictured form. You can also use a mixed approach, with each row broken into a different number of columns.

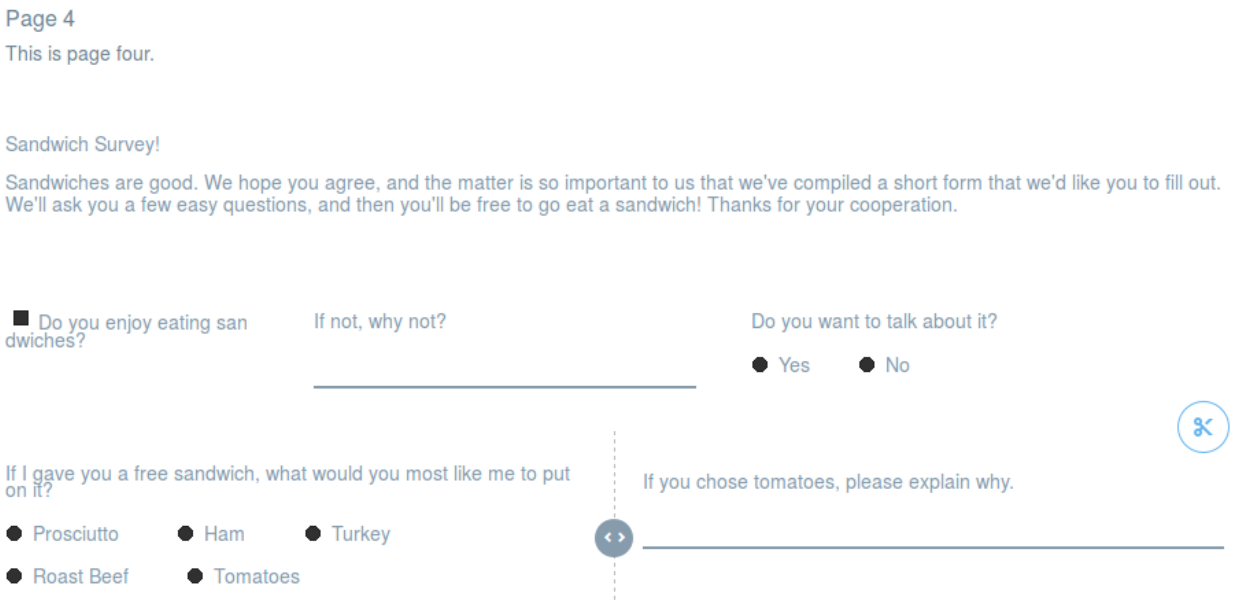


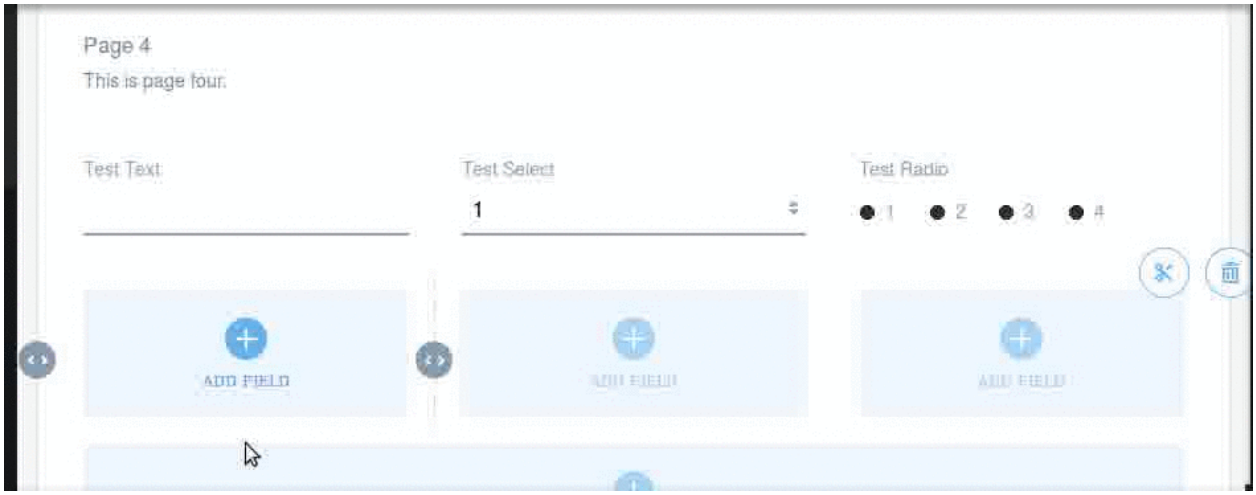
Figure 23.28: : There are many ways to lay out your Liferay forms.

If you're going to lay out your forms, you'll be working with two entities in the form builder interface: Rows and Field Slots. You've already added and configured fields. All you're doing here is learning to resize them and move them around. Rows are just that, the rows of the form.

If you've added a field to the form, you've also added a row. By default, clicking the large *Add Field* button on the form will add a field that occupies the entire row.

Figure 23.29: : By default, fields in your form will occupy an entire row.

If you'd like, you can design your forms before you add the fields. Just start resizing the row using the form layout functionality (drag the chevrons to resize the field slots, or columns, in a row). As you make a field slot smaller, the new space is taken up by another field slot. You'll end by having divided the row into two columns. Resizing again will make three columns. Once you have the row situated how you'd like, start adding the form fields.



You can also cut entire rows from the form and paste them elsewhere. Any fields you added and configured will be pasted with the field. Once you click the cut icon (the scissors), you'll be alerted to the available positions where you can paste the row and its fields.

Did you notice the delete icon ( )? Clicking it deletes the entire row, with all its fields.

Just like you can cut and paste an entire row, you can move an individual field to any unoccupied field slot, in any row. Just open the field's actions menu by clicking the Actions button ( ), which will appear when you hover your mouse over the field in the layout.

In addition to moving the field, you can edit the field or delete it.

Note: If you're following along with the employment application form, it's time to build the second page of the form (Very Personal Details). Combining your form building knowledge with your ability to customize the form layout, build the form to match the screenshot below. It will use form text, date, single selection, text, and single checkbox (multiple selection on Liferay Portal CE) fields, laid out in two columns.

Now it's done. Your users, including Guest users, can submit applications to The Lunar Resort. The form is multi-page, full of features, and there's even some processing that's done on form submission. As you can see, there's a lot of flexibility built in to Liferay's Forms application.

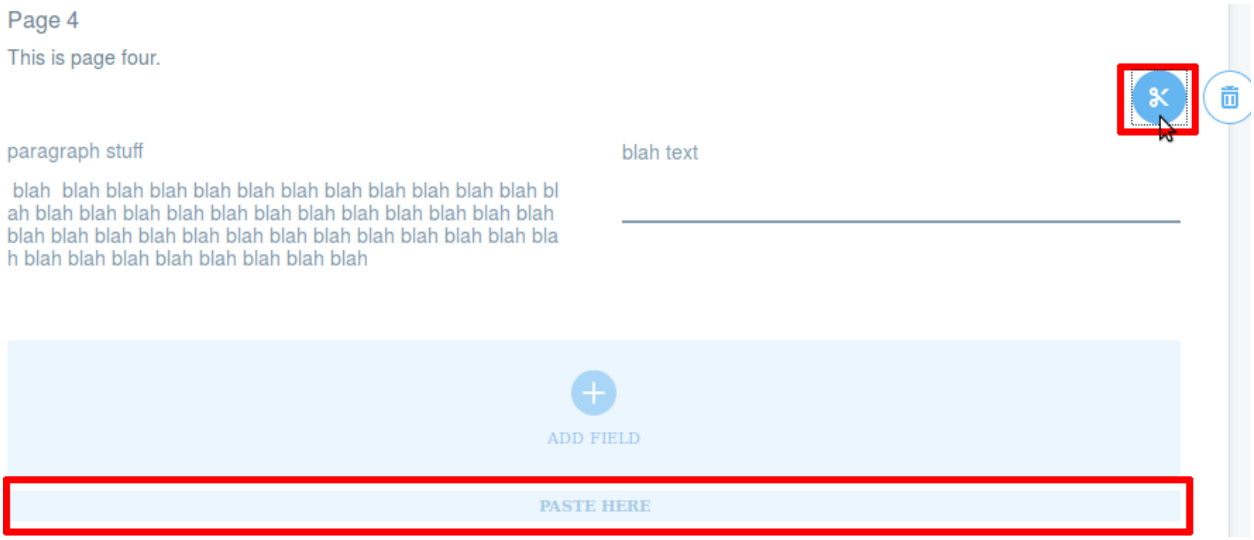


Figure 23.30: : Cut and past an entire row, including its fields, in your form layout.

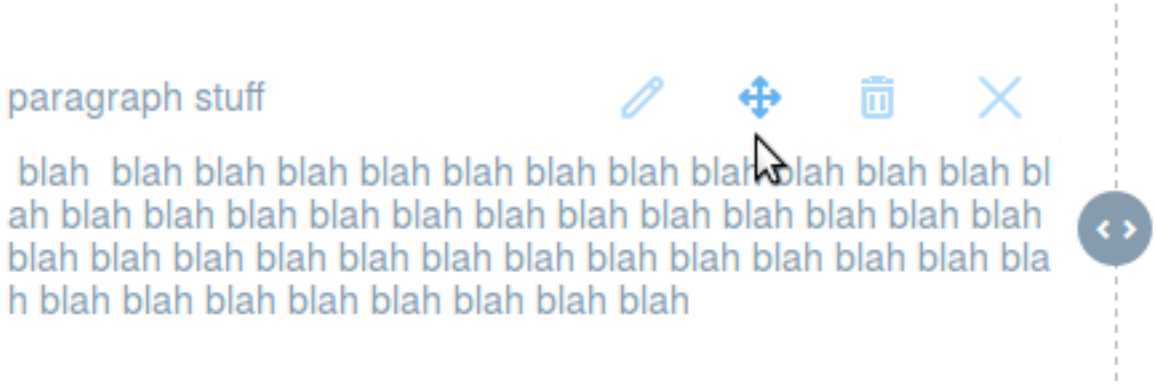


Figure 23.31: : Move an individual field in your form layout.

23.3 Sending Form Entries through a Workflow

Liferay DXP has a workflow feature that directs a submitted asset through a workflow process before it's published. Most assets are configured to use workflow at the instance or site level.

Forms is different, and that's why its assets don't appear in the above image. There are so many use cases for forms, and there could be so many per site, that a site- or instance-scoped workflow configuration won't serve your needs well. Instead, the Forms application lets you configure the workflow that's used for *each form*. That means that each submitted form record will go through a workflow process before being marked as *Approved*. Configure workflow for a form in its Settings window.

Enabling Workflow in a Form

To enable workflow in a form:

- Open the form's editor by navigating to *Site Administration* → *Content* → *Forms*, and clicking on the form you're interested in.

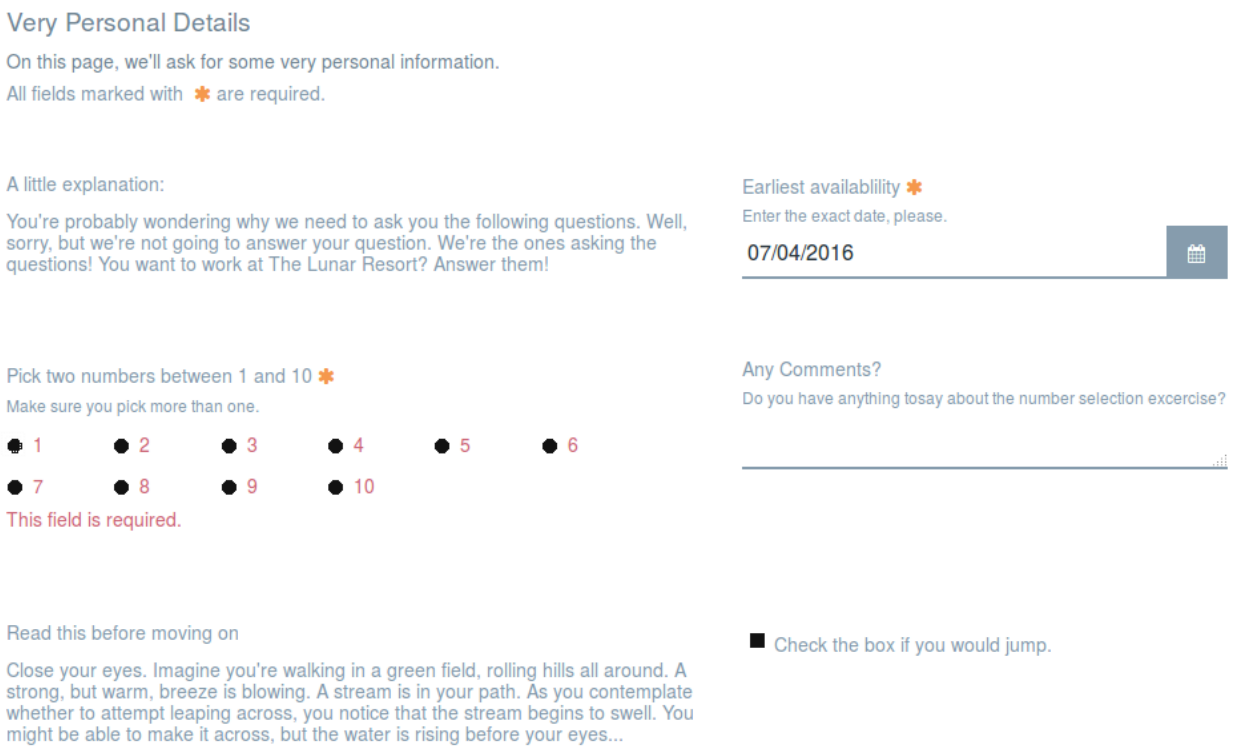


Figure 23.32: : The second page of The Lunar Resort employment application.

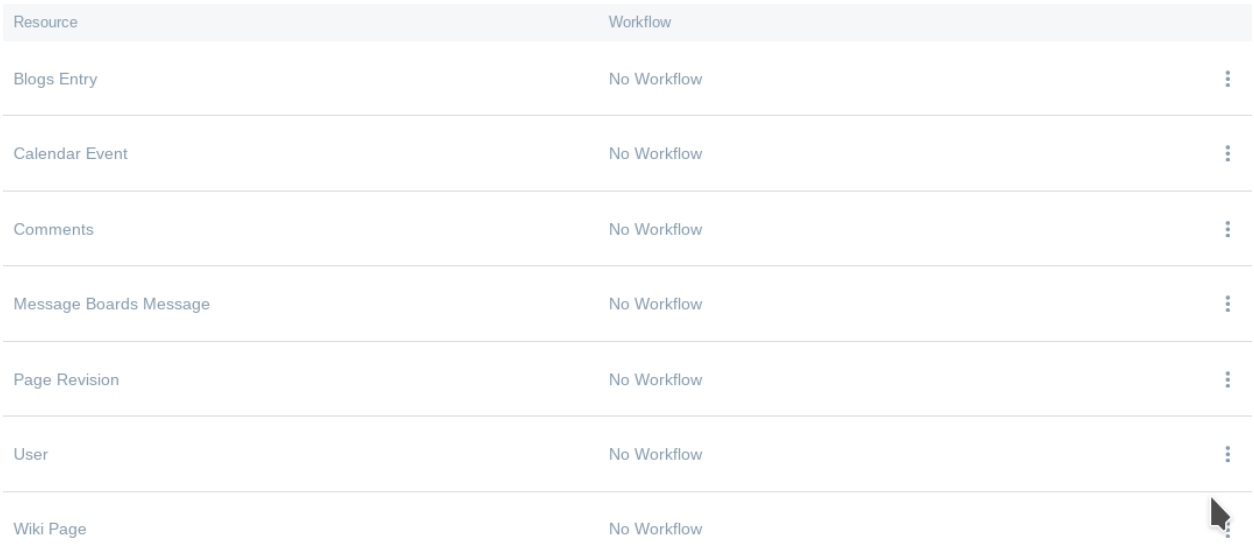


Figure 23.33: Workflow is enabled in the Control Panel or in Site Administration for most assets.

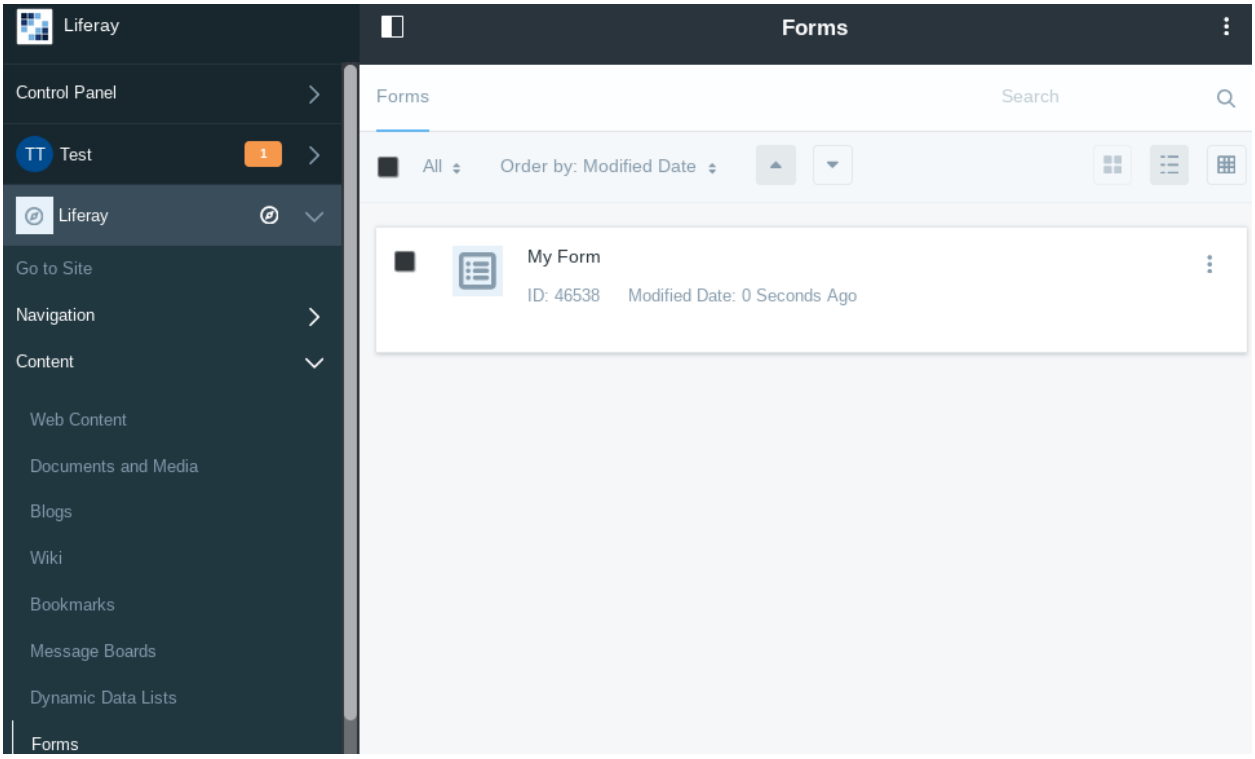


Figure 23.34: Navigate directly to a form to enable workflow.

- Click the Options button ( ) and choose *Settings*.
- The Settings window has a *Select a Workflow* dropdown. Find the workflow you're interested in, select it, and then click *Done*.

Testing the Workflow

Test the workflow process by submitting a form entry. If you're using the Single Approver workflow that ships with Liferay DXP and you're the administrative user, you'll see a notification come in after submitting the form entry, alerting you that there's a form record to review.

Next go find the form entry in the Forms application:

- Go back to the Forms application in Site Administration.
- Click the Form's Actions button ( ) and select *View Entries*.

The entry is currently marked *Pending*. Now approve the form record:

- Navigate to *My Account* → *My Workflow tasks*.
- Click the *Assigned to My Roles* tab.
- Click on the form entry.
- Click the Actions button ( ) and choose *Assign to Me*.
- Click *Done*.
- Click the Actions button ( ) again, then click *Approve*.
- Click *Done* again.

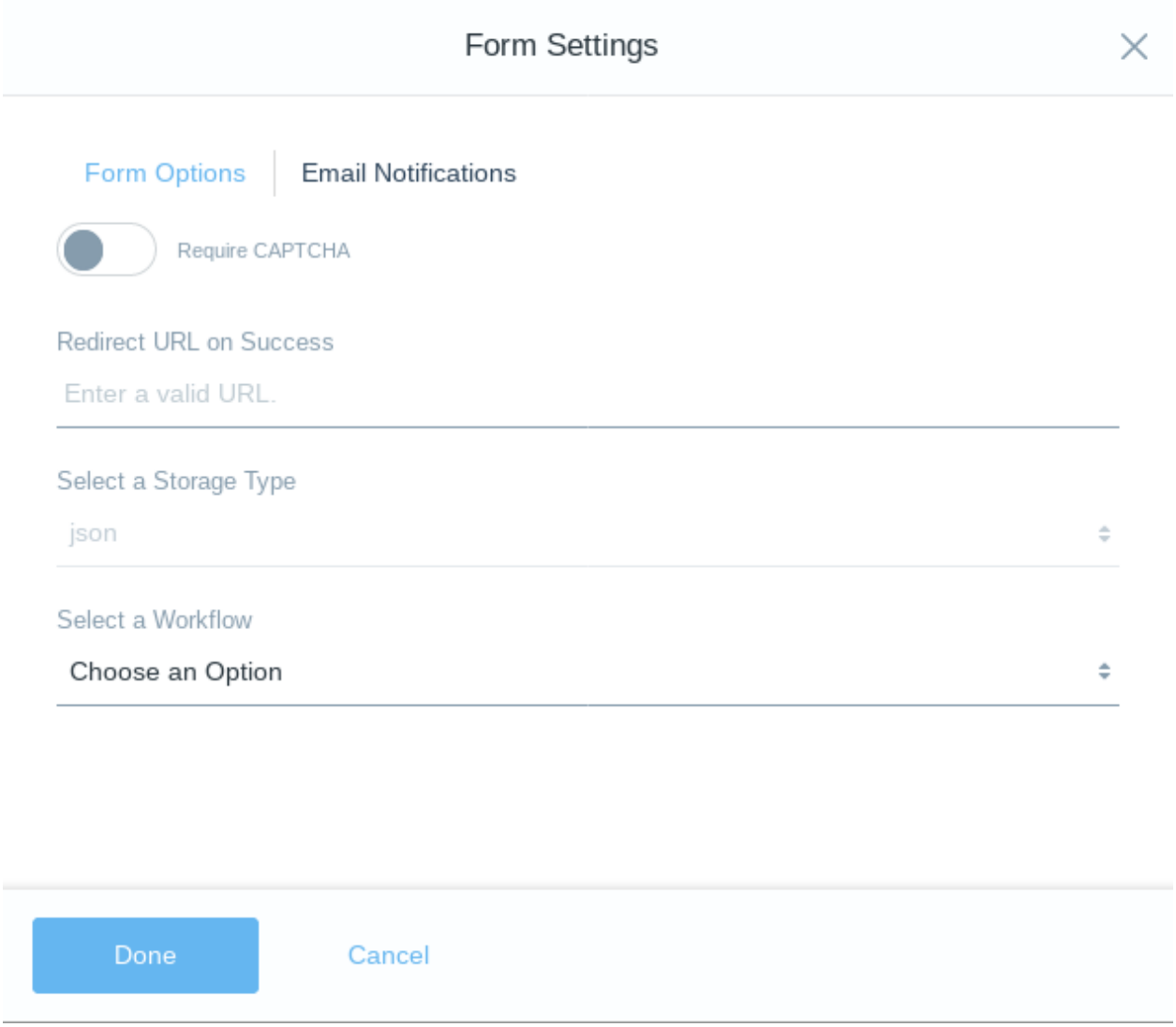


Figure 23.35: Enable workflow for each form in its Settings window.

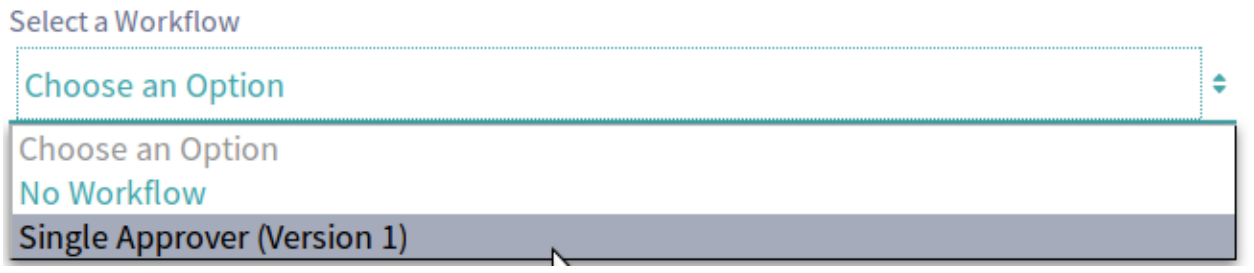


Figure 23.36: Select the form's workflow using the Select a Workflow dropdown menu.

Navigate back to the View Entries screen for the form, and now the entry is marked as *Approved*.

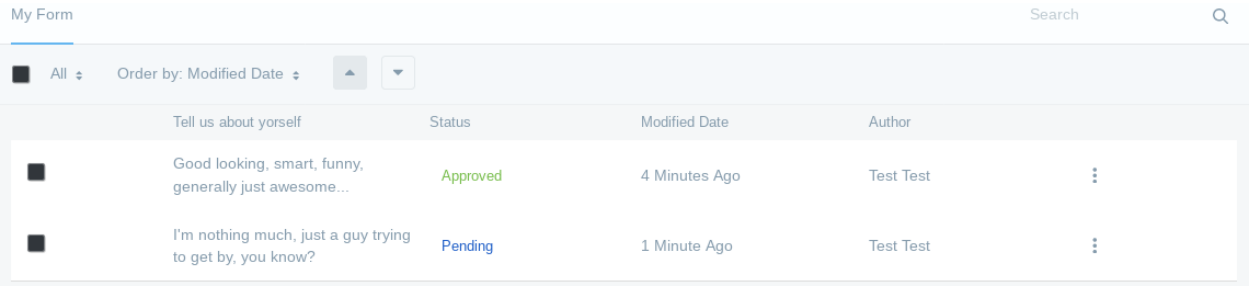


Figure 23.37: Each entry's status is visible in the Forms application's View Entries screen.

Example: Win a Free, All-Inclusive Trip to the Lunar Resort

The Lunar Resort is giving away a free, all-inclusive trip to several lucky families. All a user must do is click the *I Want a Free Trip* link from the site homepage, and they're directed to a form they can fill out and submit in under three minutes:

The Lunar Resort wants to exclude folks who don't meet certain qualifications:

- Exclude families with more than five persons in the household.
- Prioritize families who've never visited The Lunar resort before.
- Exclude anyone who's on the resort's naughty list, appropriately titled "The Dark Side of the Moon".

To compile a list of finalists, each form entry should go through a workflow with one review task. The Single Approver workflow that ships with Liferay DXP works nicely for this.

After creating the form in the above image, select the Single Approver workflow from the form's Settings window, as described in the above section. Publish the form, then fill out and submit an entry. The administrative user receives a notification that a form record was submitted. Go to the View Entries screen and see that the form record is currently marked as Pending. Once the review is completed, it's marked as Approved.

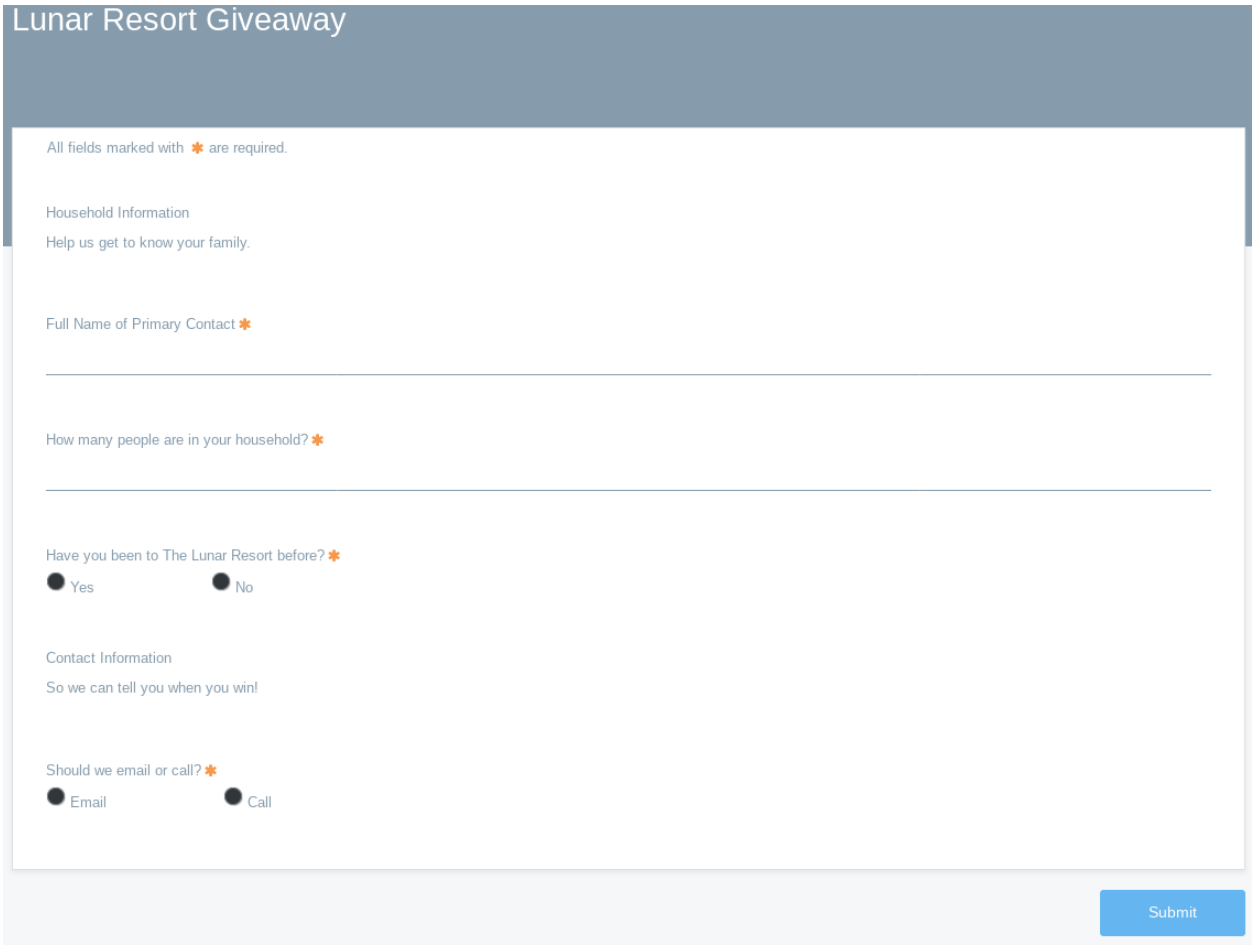


Figure 23.38: The Lunar Resort Giveaway form is ready to be filled out.

Chapter 24

Creating Simple Applications

Business needs change, and technology must adapt to meet those needs. Things that used to be done using manual processes are now done electronically. You may want your team to sign up on your web site for a holiday party. Maybe you need to put up a job posting board every fall, where only administrators can create new job posts. Maybe you'd like users to manage a notebook or To-Do list on their private pages. In these cases, you need custom sets of data, and your users need to add and access their information.

In the past, you'd need to be a developer to accomplish any of this. Today, you can do it without writing a single line of code. Liferay's *Dynamic Data Lists* application provides an easy way to create, aggregate, and display new data types. Data Lists are flexible enough to handle all types of data, and you don't have to write any code. Simply put, Liferay gives you the power to perform the following actions:

- Create new lists from data definitions you build yourself
- Create new lists from predefined definitions
- Customize the input forms for ease of use
- Customize the output format

All of this capability is summarized in two concepts: data defining and data displaying. These data lists are dynamic for a reason: they are flexible and powerful. From collecting simple input from the user ("Did you find what you were looking for?") to developing an entire data entry system for real estate listings, Dynamic Data Lists have your use case covered. Combined with the flexibility provided through templates and the power of languages like FreeMarker, entire applications can be built in a short time.

24.1 Creating Data Definitions

Imagine this: The Lunar Resort is becoming a popular destination. The Recreation Department's offerings (hiking, lunar golf, rover races, and more) are very popular with the adventurous guests, and it's becoming difficult for Recreation Department employees to keep up with demand. An unexpected number of participants in a lunar hike makes it so that the Lunar Sherpa (hiking guide) has to scramble to find enough space suits for all the guests involved, and the kitchen staff needs to rummage up more food at the last minute so that the hikers have enough to eat. The lack of efficiency could be dangerous, and cost the resort much needed revenue if guests are unhappy with the experience. The resort needs a way for guests to sign up for

events ahead of time. If you're thinking that means you need to hire developers to build you an app, you're wrong! Liferay gives you, the administrator (or anyone with the proper permissions), the power to build a simple list app using the Dynamic Data Lists application, with no coding involved.

The Lunar Resort has a public site for guests. On that site, there's a page called *Activities Sign-Up*, where guests can sign up for activities led by the Recreation Department, or for the upcoming Lunar Luau, a Hawaiian themed dinner party on the Moon! It's important to know which guests are attending which activities, since any lunar activity requires special equipment, considerations, and preparation. You need to do two things:

1. Provide guests with sign-up forms.
2. Access the guest lists to view the entries.

These are basically simple applications, that take user input and store the data somewhere. While you don't need to be a developer to make this happen, you do need to be familiar with data definitions, data lists, and templates. You probably also need a user group of List Creators who have permission to create lists and access the entries.

When list creators sign in to the Lunar Resort portal, you want to make it easy for them to build lists. To this end, you'll provide one or more data definitions that should suit most of their list making needs. Not everyone needs to be an expert in Dynamic Data Lists (but you do, so keep reading).

What are these data definitions? They're just collections of different field types, configured in a certain way, to guide users in providing data for your lists. All lists are backed by a data definition, so this is the place to start in building a list app. An exhaustive discussion of the field types available and their configuration options is found at the end of this article. Next, get started creating data definitions.

Creating Data Definitions

The Lunar Resort List creators can't post and manage lists unless you give them some useful definitions. You can, of course, delegate the data definition responsibilities to the list creators, if you want. The following directions apply to whoever has the permissions to access the DDL application. Create some data definitions and set your list creators to work.

1. Make sure you're logged in as an administrator, then open the site administration menu for the site you're working on. Click *Content* → *Dynamic Data Lists*.

2. To create a data definition, find the icon ( ) for the Dynamic Data List configuration menu. Click on it and select *Manage Data Definitions*. The Data Definitions window appears, where you can access existing definitions or create new ones.

3. Click the *Add* icon ( ) to add a new Data Definition.

4. Fill out the Title and Description fields. Since this definition will be used for multiple lists (because there are multiple activities the resort needs lists for), name it *Activity Entry*. Beware that the Title you give the data definition will appear to portal users who are adding a record to your list. There will be a button that users click to *Add a [Definition Title]*. A good description might be *This definition should cover most use cases on the Activities Sign-Up page. Modify the templates to suit your specific needs.*

   Below the Description field, there's a Parent Data Definition selector. The Activity Entry definition won't use any of the pre-configured definitions as a parent, so leave it blank and scroll down to the data definition designer. The *View* tab is displayed by default, and that's where you'll probably do most of your work. You can, however, click *Source* mode to manually customizing a definition's XML file. This method is likely to be useful for more experienced users and developers.

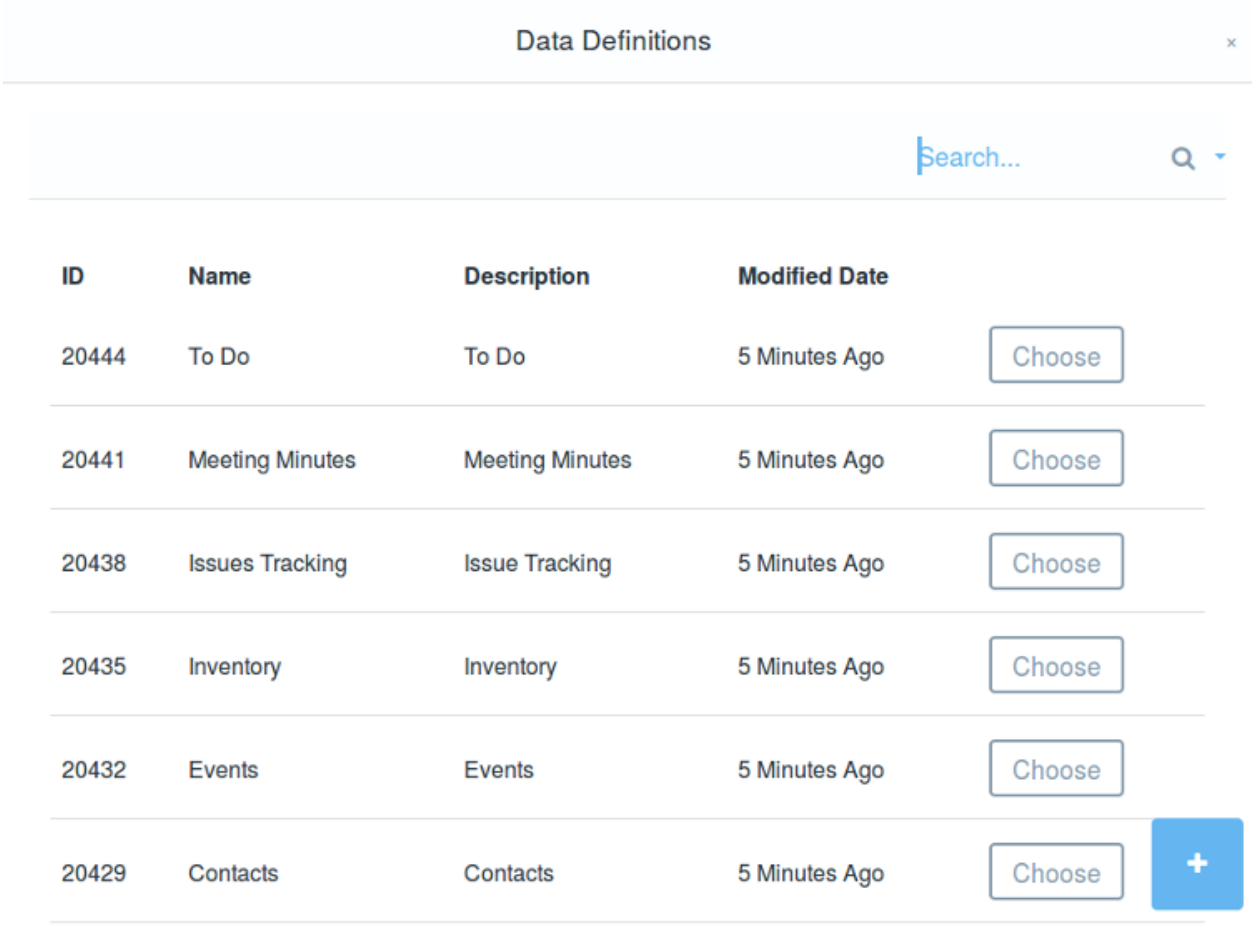


Figure 24.1: The portal ships with several commonly used predefined data definitions that can be used as is or modified to build your lists. You can also add your own data definitions.

5. To start, drag a Text field from the Fields tab (in View mode) onto the canvas. This will eventually become a required field called Full Name. Below the Text field, drop a Radio field. Drop four Boolean fields, and finish this section with a Select field. While doing this you might have noticed you can nest fields below each other. Nest all the fields you just added below the Text field.

```
**NOTE:** When you nest fields, they will not appear when the list entries
are displayed with the default display template. For more information about
templates, see the article on
[Using Templates to Display Forms and Lists](/docs/7-0/user/-/knowledge_base/u/using-templates-to-display-forms-and-lists).
```

---

6. Before adding more fields, configure the current ones. Click on the Text field and then click on the wrench icon ( ). Alternatively, click on the field and then click the Settings tab.

    - Double click the Field Label and change it to *Full Name*. Click Save.

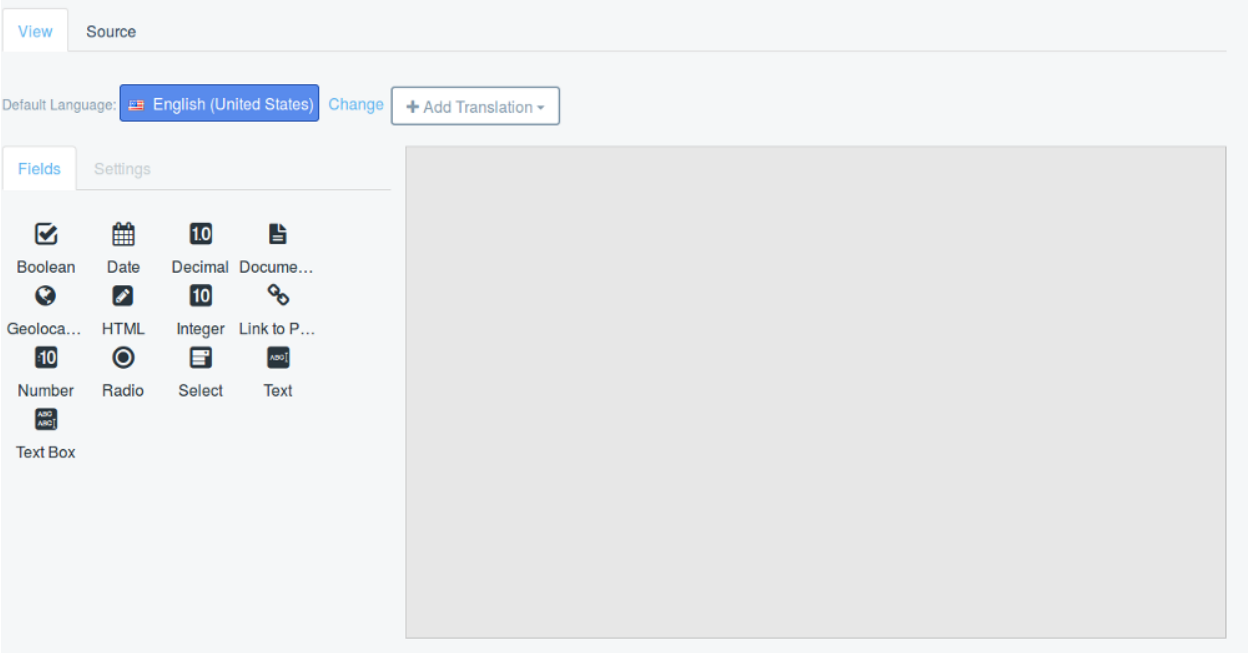


Figure 24.2: You can add and configure fields to form a custom list definition.

- Double click the Required setting and change it to *Yes*, then click Save.

7. Next, configure the Radio field.

   - Change the Field Label to *Sex*.
   - Configure the field Options to be *Male* and *Female*, and feel free to set the values for the options to *male* and *female*. Only the Options will be displayed to users. Remove the unused Option.

8. Now create and configure DDL data definition fields to make the definition look like this image:

```
![ You can add and configure fields to form a custom list definition.](./images/ddl-name-and-more.png)

-  The options for the Select field  wtih the label *How fit are you?* should be *Fit as a
  fiddle!*, *Moderately fit, I think.*, and *Bring extra oxygen.*
```

13. That's a good start to the form. To make sure all Activity Sign-Up use cases are covered, add the following fields to the data definition:

```
-  Documents and Media field called *Profile Picture*.
-  A Radio field called *Preferred Rover Style* with these options:
    -  *Uranus (light blue)*
    -  *Jupiter (orange and white)*
    -  *Mars (red)*
-  A Geolocation field with the label *Tell other guests where you're from.
   Optional.*
-  A Text Box field with the label *Tell us something about yourself.*
```

14. Once you finish adding and configuring fields, click Save.

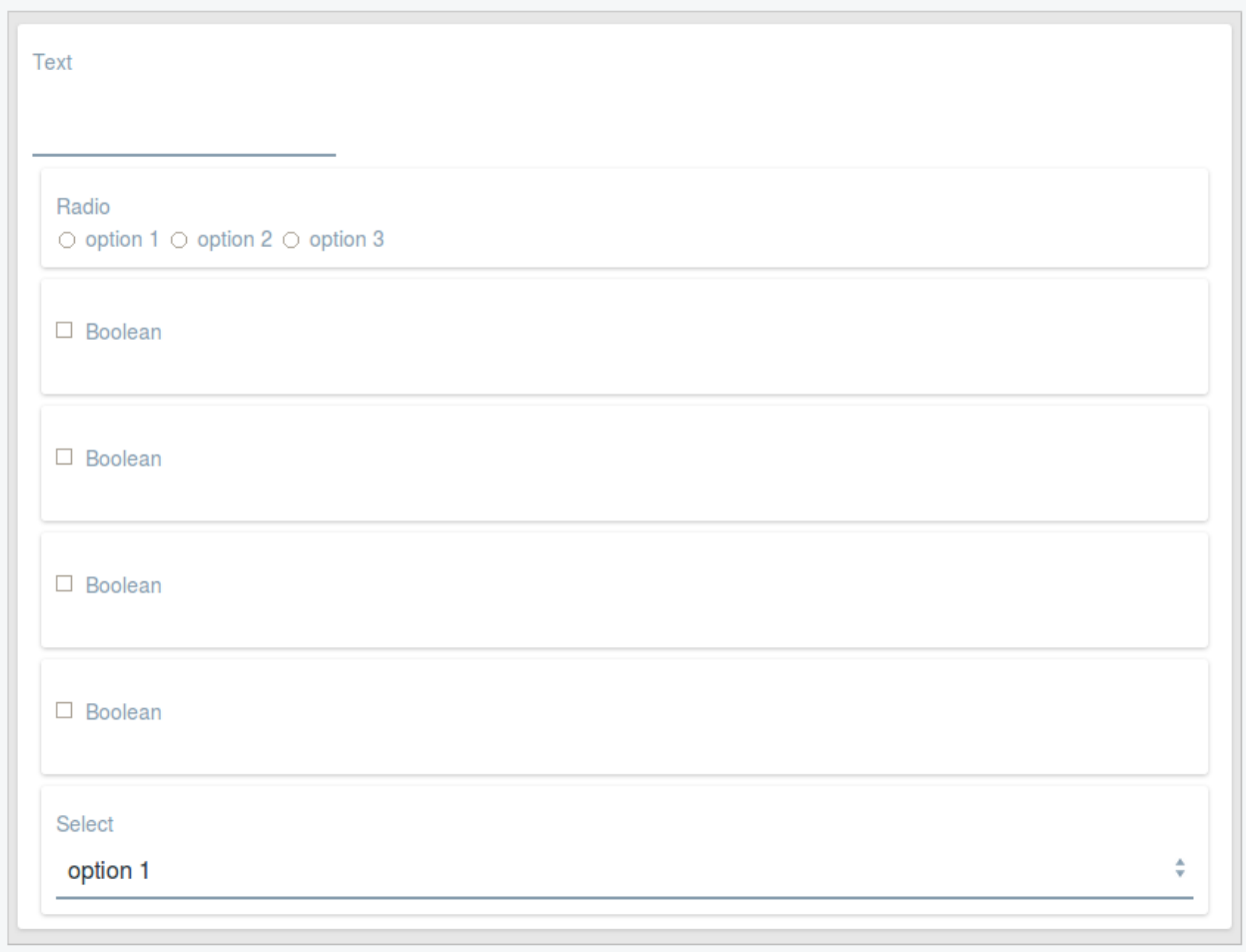


Figure 24.3: Fields can be nested below other fields.

Your new data definition will now appear in a list with all of the pre-configured definitions.

If you click on the newly created definition (or any definition, for that matter), you'll notice that you have access to both a WebDAV URL and a static URL, which are used to access the XML source of the data definition. To learn more about WebDAV or if you'd like to see WebDAV in action, see the article on WebDAV access .

Using a Parent Definition Though it wasn't necessary for the use case presented above, data definitions also have the capability of inheriting characteristics from other definitions. When a parent data definition is configured, the child definition inherits the parent's fields and settings. Using this feature is helpful when you want to make a similar data definition to one that's already created. For instance, if you'd like to create an advanced sign-up sheet in addition to a regular sign-up sheet, you can simply inherit the characteristics of the regular sheet and only add the additional fields necessary for the advanced sheet. When the advanced sheet is configured, it will display its parent's fields in addition to its own fields.

If you wanted the Sales Department to fill out a form to help keep inventory of the Lunar Resort store's items, you could build a list based on the Inventory Definition that comes shipped with Liferay. It has some useful fields already configured, but the existing Inventory data definition isn't enough. You want to give Lunar Resort employees a chance to rate the popularity of each item they're filling out the Inventory form for.

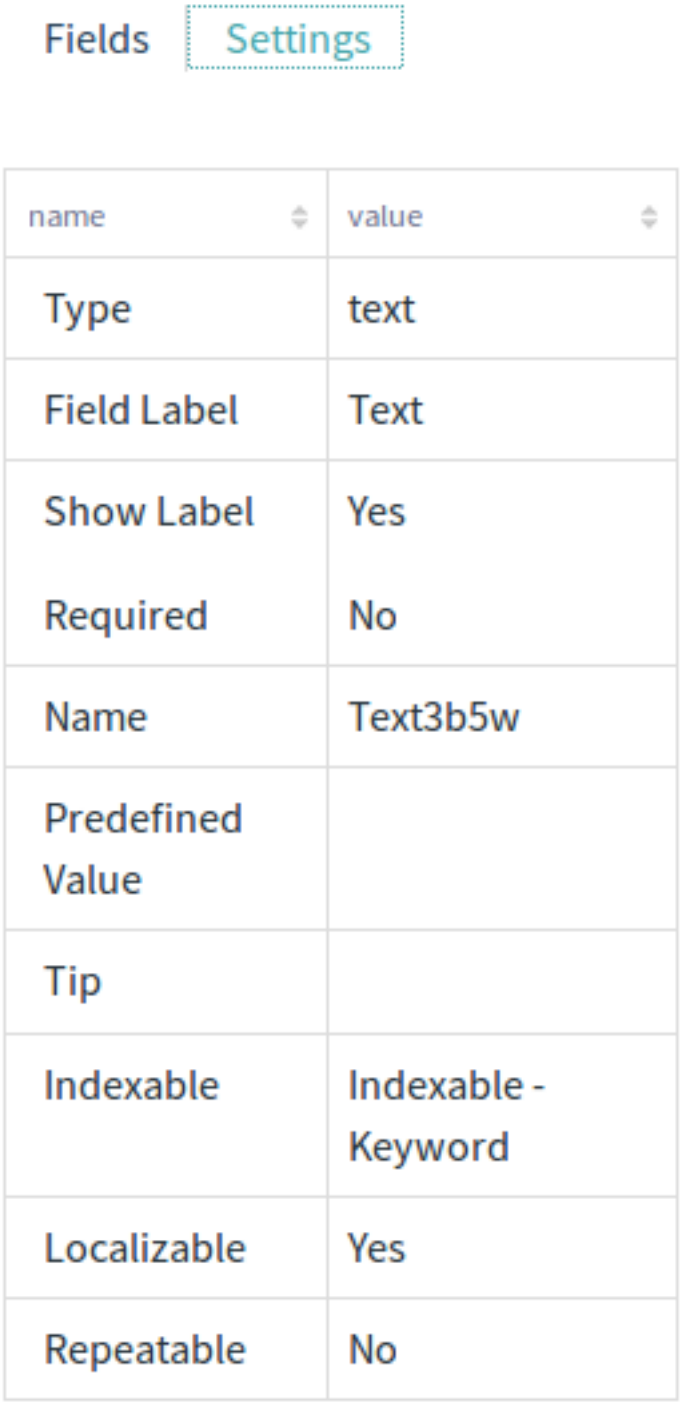


Figure 24.4: Fields can be configured in a variety of ways.

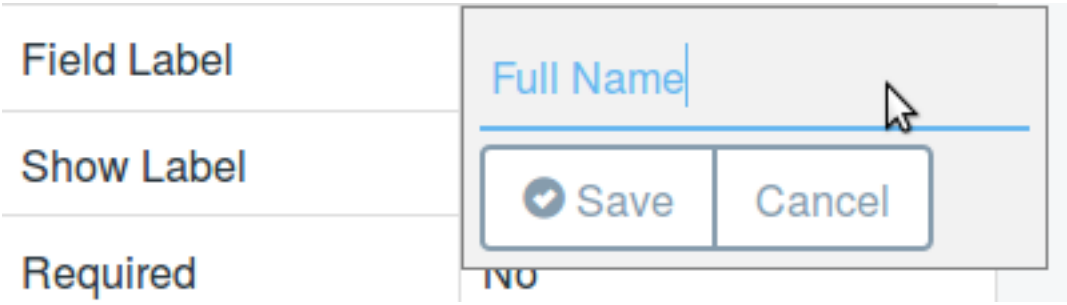


Figure 24.5: You can give your fields appropriate field labels.

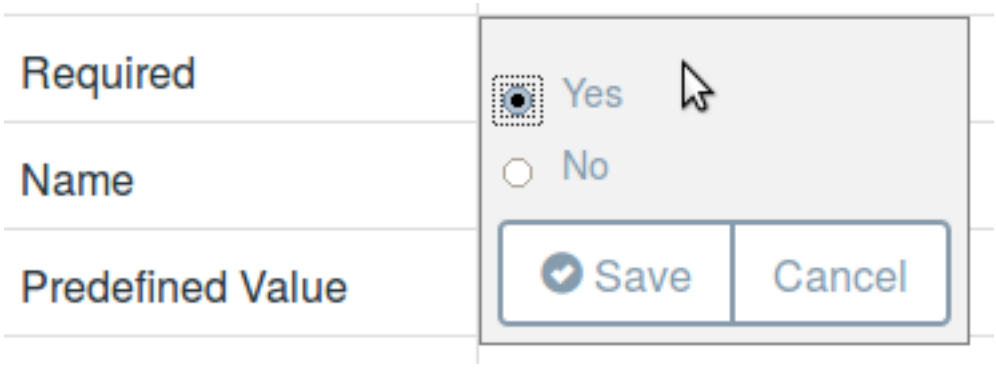


Figure 24.6: You can make your fields required.

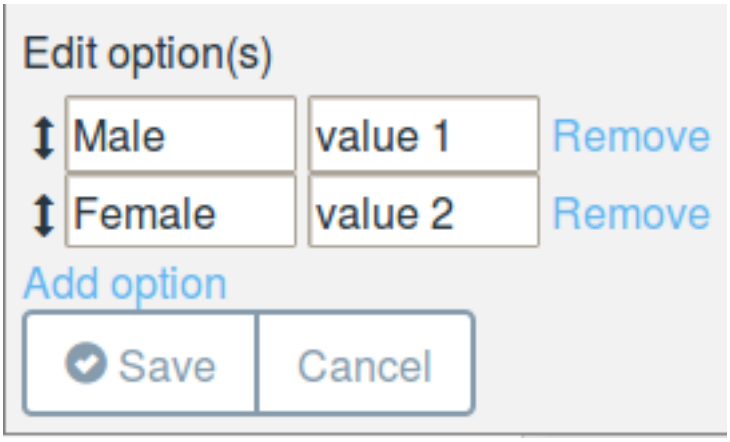
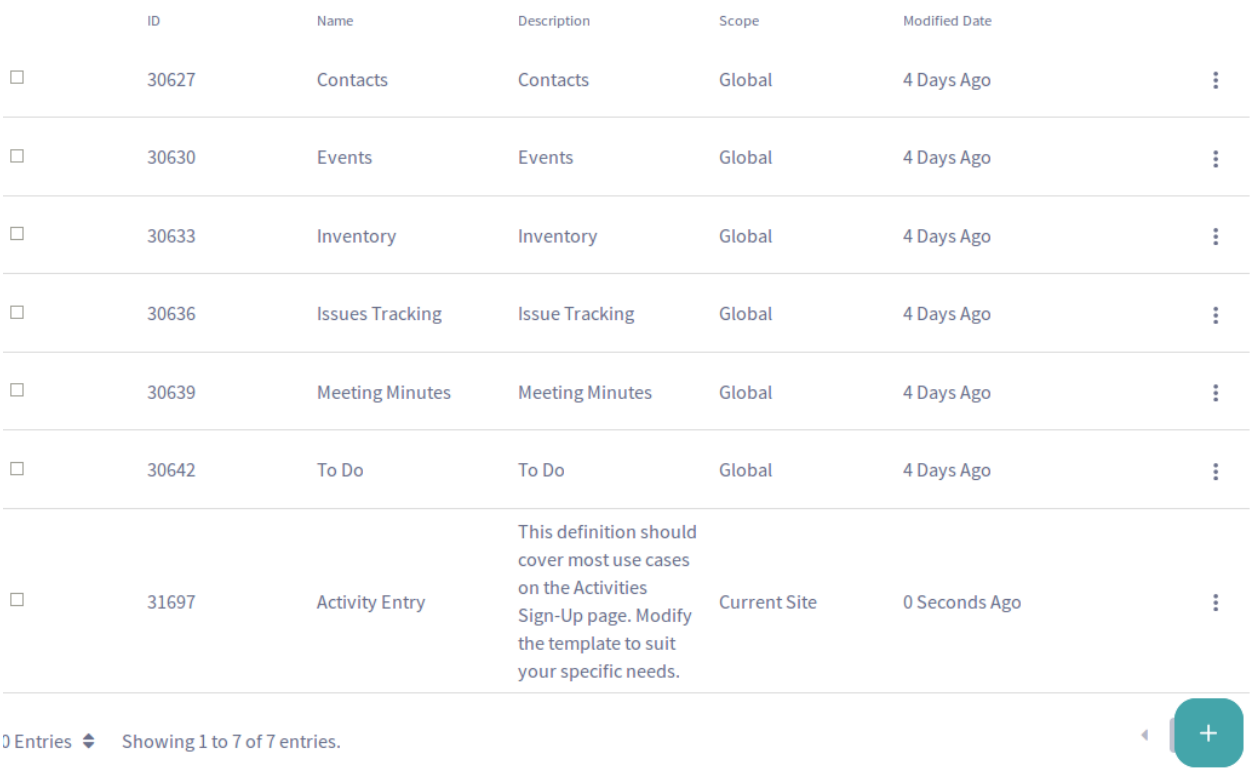


Figure 24.7: You can add and configure fields to form a custom list definition.



	ID	Name	Description	Scope	Modified Date	
☐	30627	Contacts	Contacts	Global	4 Days Ago	⋮
☐	30630	Events	Events	Global	4 Days Ago	⋮
☐	30633	Inventory	Inventory	Global	4 Days Ago	⋮
☐	30636	Issues Tracking	Issue Tracking	Global	4 Days Ago	⋮
☐	30639	Meeting Minutes	Meeting Minutes	Global	4 Days Ago	⋮
☐	30642	To Do	To Do	Global	4 Days Ago	⋮
☐	31697	Activity Entry	This definition should cover most use cases on the Activities Sign-Up page. Modify the template to suit your specific needs.	Current Site	0 Seconds Ago	⋮

Figure 24.8: You can add and configure fields to form a custom list definition.

Just create a new definition, define the Inventory definition as a Parent, and add the fields you need.

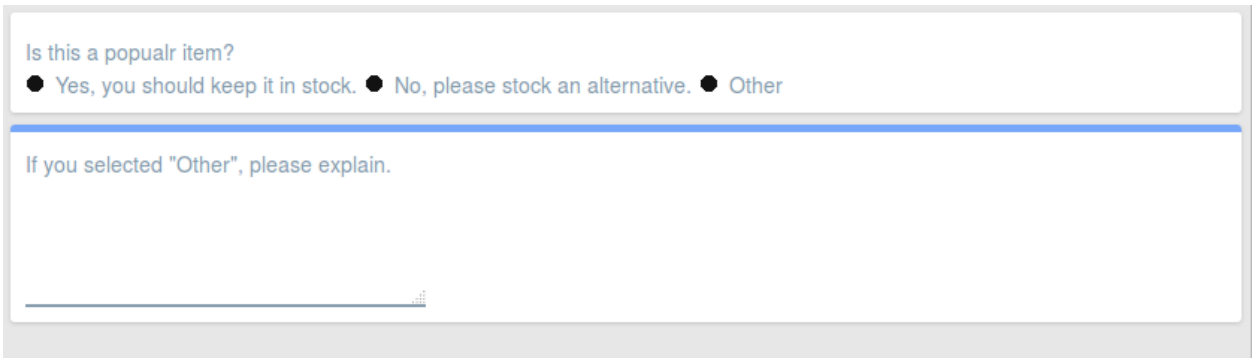
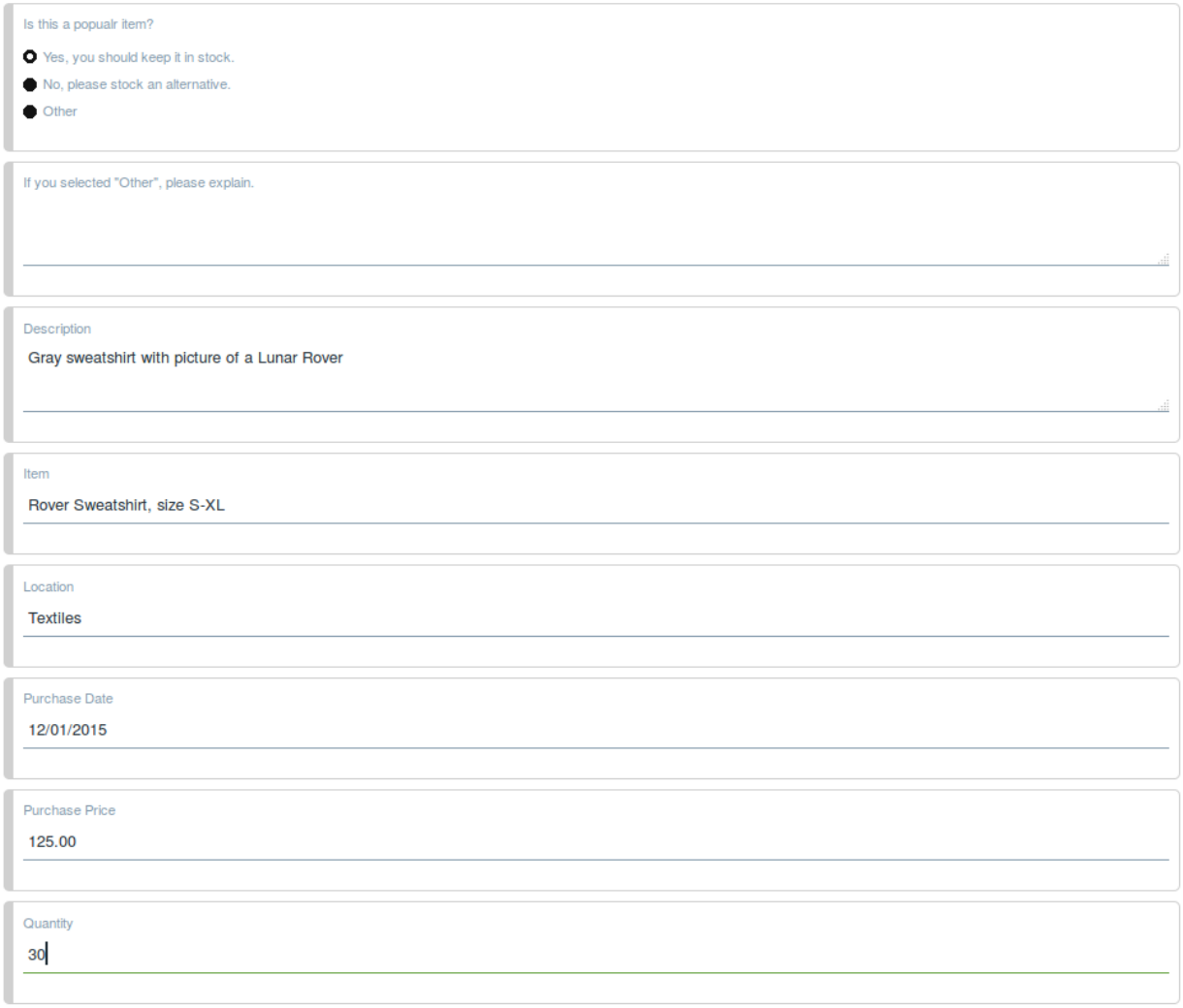


Figure 24.9: : You can create a child definition to add new fields to a pre-existing one.

When you create a list (you can call it Inventory List) based on the new definition (probably called something like Inventory Record), it will include the fields you added and the fields of the existing definition. This way, whoever is in charge of maintaining the Inventory definition gets the information they want, and you get the information you want.



Your definition now lives with all the pre-configured ones. There are several things you can do with these existing definitions, including deleting them, editing them, and copying them.

Managing Data Definitions

From a site's Dynamic Data Lists application (accessed through the Product Menu), you can either add a new dynamic data list from an existing data type or you can click *Manage Data Definitions* from the Actions button to add or edit data definitions.

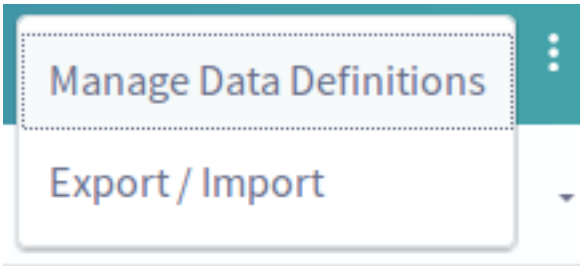


Figure 24.10: : You can manage the portal's data definitions.

There are several data definition management actions you can perform. Of course, you can edit a data definition, but you can also configure its permissions, manage its templates, copy it, or delete it.

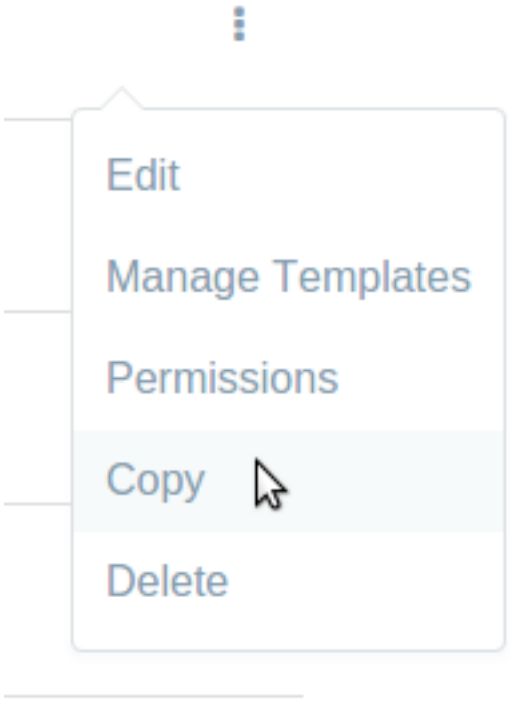


Figure 24.11: : You can copy an existing data definition, manage its templates, and more.

If you edit a data definition that has references (from your lists or templates), you'll need to account for that, by editing the template, for instance.

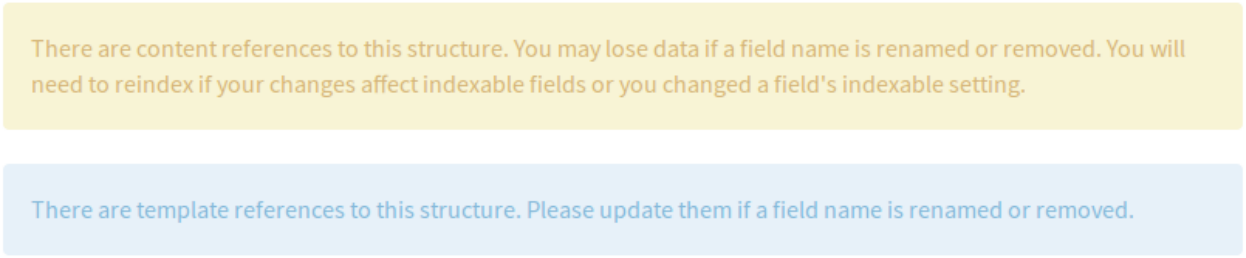


Figure 24.12: : You'll be warned if you try to edit a data definition that's referenced in a template.

Copying Data Definitions Liferay Portal 6.2 introduced the ability to *Copy* the DDM structures and templates associated with an existing data definition. You can access the Copy button by navigating to *Manage Data Definitions* and clicking the *Actions* button (the icon is a vertical ellipsis). Select *Copy* from the actions menu. The Copy menu opens, and includes options for copying the form and display templates associated with the data definition. You'll learn how to manage and create new form and display templates later. When you're finished, the copied data definition can be accessed in the *Manage Data Definitions* menu. The Copy feature lets you create new data definitions based on existing ones. In one action you can copy a data definition

with its templates, and then you can modify it to suit your needs. You can, of course, edit any definition in the portal, but if you copy a definition instead, you'll still have access to the original.

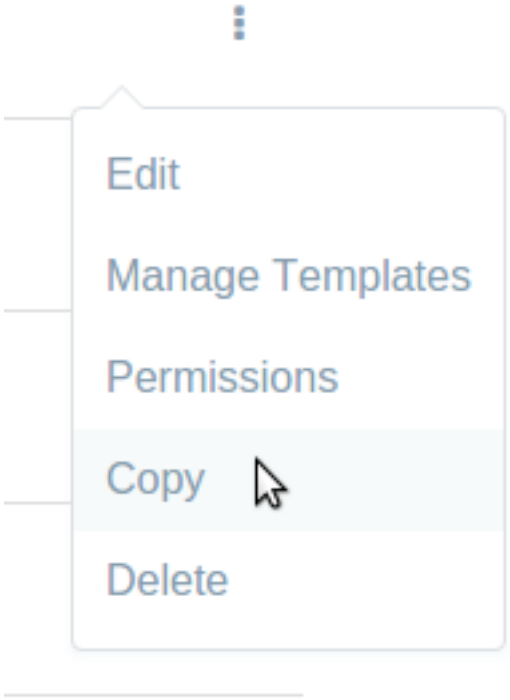


Figure 24.13: : You can copy an existing data definition's DDM structures and templates.

From the actions menu for a data definition, you can also configure its permissions.

Data Definition Permissions From a data definition's Actions menu, you can select Permissions to configure the permissions for a particular data definition. Note that these permissions are for an individual definition accessed through the DDL application in Site Administration (from the Product Menu, go to *Sites → Content → Dynamic Data Lists*). For example, by default a Site Member can View the Activity Entry Data Definition. Any Site Member who was also assigned a role that can access the DDL application and its data definitions would see this definition listed in the Manage Data Definitions list. If you don't want this, just deselect View for Site Member, and these users won't see your data definition listed with the others.

You can manage a data definition's templates as well.

Managing a Data Definition's Templates Manage a data definition's templates by clicking Manage Templates from a data definition's actions menu (click the icon).

You can copy templates if you want to create a new template that's similar to an existing one, or you can edit them directly. You can also configure permissions for a template, and delete templates. All of these are accomplished by clicking the actions button for a template. For more information see the article on Using Templates to Display Forms and Lists.

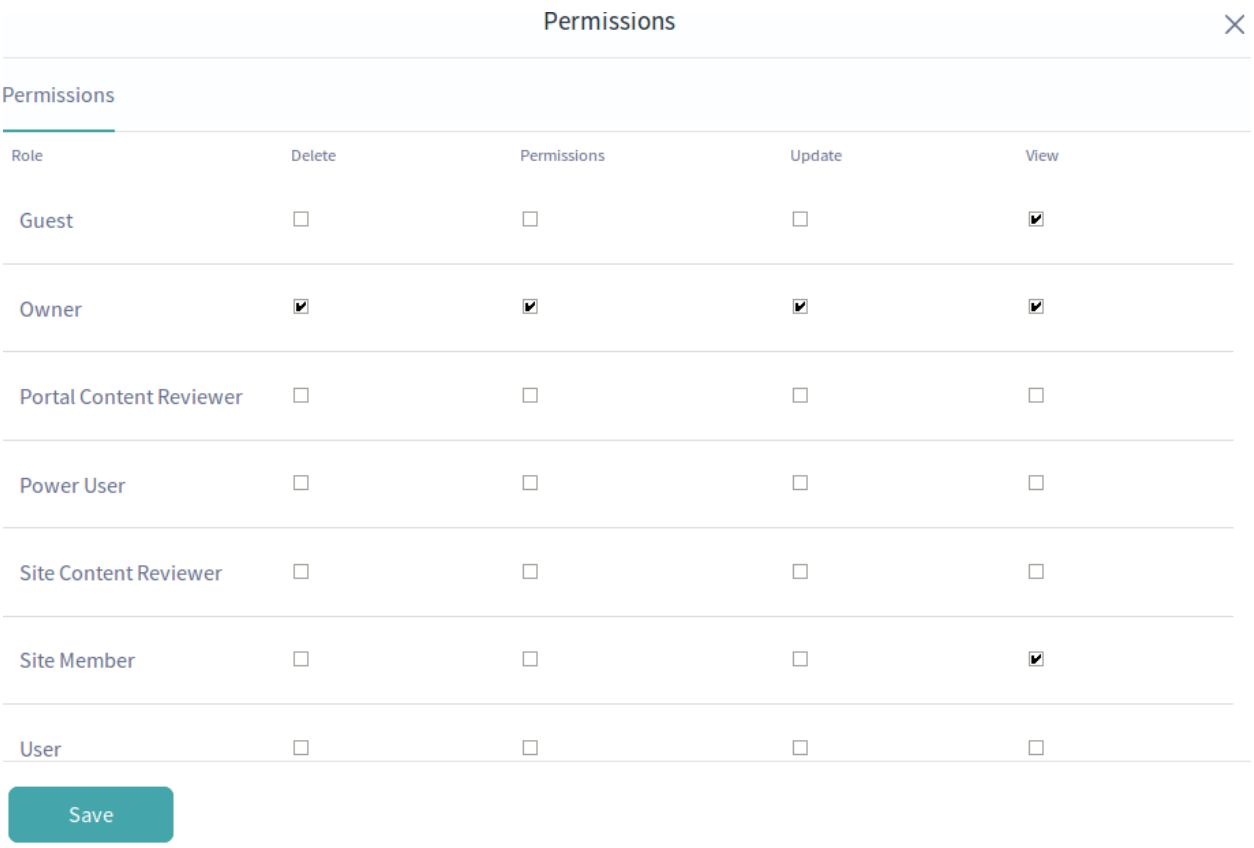


Figure 24.14: : You can configure the permissions on an existing data definition.

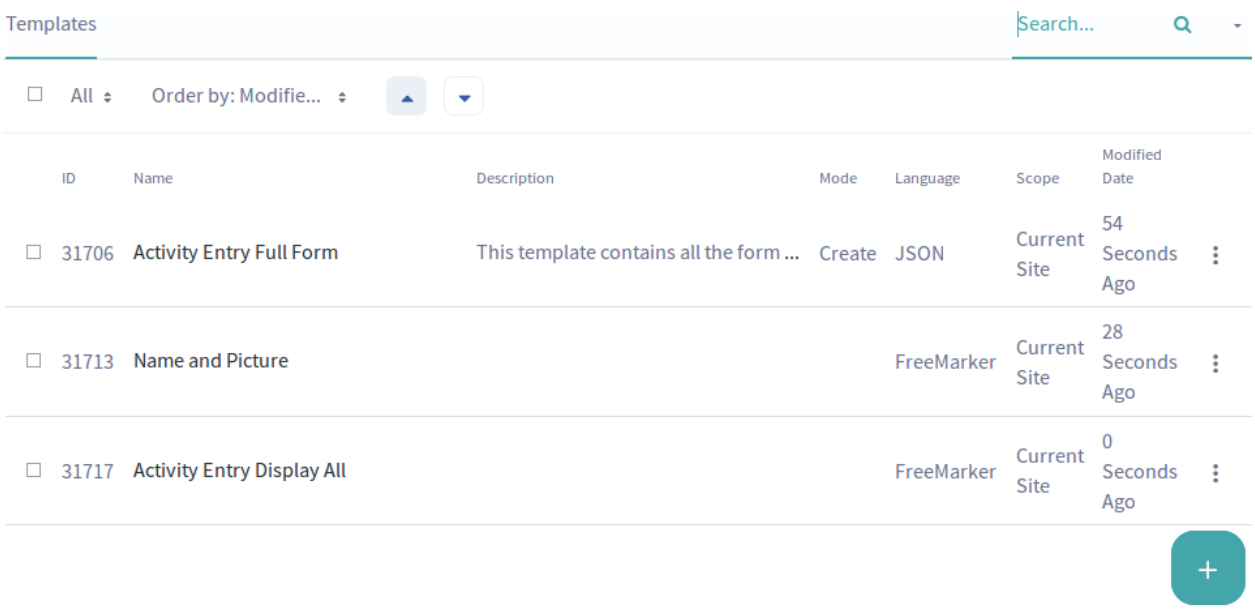


Figure 24.15: : You can manage the templates associated with a data definition.

Configuring Data Definition Fields

You already know how to create data definitions, and if you created the example definition, you've added and configured some of the field types that can be used in your definitions. But there's more to know about the field types and their configuration options.

There are many different types of fields to choose from when creating data definitions:

Boolean fields present a checkbox to the user and stores either a `true` (checked) or `false` (unchecked) based on state.

Date fields are preformatted text fields that display a convenient date picker to assist in selecting the desired date. The format for the date is governed by the current locale.

Decimal fields are similar to Number fields, but require that a decimal point be present.

Documents and Media fields allow users to select an existing uploaded document to attach to the data record. Users can also upload documents into the Document Library.

Geolocation fields are integrated with Google Maps, and allow users to enter an address, which is then displayed on a map.

HTML fields present an area that uses a WYSIWYG editor to enhance the content.

Integer fields are similar to Number fields, but constrain user input to non-fractional numbers.

Link to Page fields allow insertion of a link to another page in the same site.

Number fields provide a text box that only allows numeric input, but put no constraints on the kind of number entered.

Radio fields presents are similar to Select fields, presenting the user with a list of options to choose from using radio button inputs. The values are stored as strings, and only one option can be chosen.

Select fields provide the user options for the user to choose from using a combo box. Unlike Radio fields, Select fields can be configured to allow multiple selections.

Text fields are areas where any string input can be entered.

Text Box fields are larger text areas for longer text input.

If you're in the Dynamic Data List's Add Data Definition section, you add these fields to a definition by dragging them from the palette on the left to the work area on the right.

You can also customize the appearance of input fields and provide helpful tips and hints for those entering data. Some data types have specific configuration options but all have some in common. Field properties can be edited in several ways: by double-clicking on the field, by clicking the wrench icon in the upper-right corner of the field,or by clicking the *Settings* tab when the field is selected:

The *Type* setting can't be edited, but can be referenced from a display template. It lists the type of field placed in the definition, and is available to reference from a display template.

Edit the *Field Label* to set the text that's displayed with the field.

The *Show Label* setting is used to configure whether the field label is shown with the form field.

Required can be set to *Yes* (it defaults to *No*), which means the field must have data in it for a new entry to be submitted (not available for Boolean).

The *Name* property sets the name of the field internally, and is automatically generated. This is the variable name that you can read the data from in a display template, so you should specify a memorable name here.

Set the *Predefined Value* to provide example data or a default value for the user to start with. The value you enter becomes the field's default value when adding a new entry.

Use a *Tip* for the field to provide helpful text that's displayed in a tooltip, which pops up from a small help icon next to the field.

Set *Indexable* to *Yes* if you want Liferay to index your field for search.

Setting *Localizable* to *No* turns off localization for a particular field, so that translations cannot be made for that field.

Setting the *Repeatable* property to *Yes* lets users add as many copies of this field as they like.

Set the visual *Width* of the form on the page. It doesn't affect the field values that will be stored. Possible width values are *Small*, *Medium* and *Large* (not available for Boolean, Documents and Media, Geolocation, Radio, and Select).

Setting *Multiple* to *Yes* allows the user to select more than one option. This defaults to *No* (only available for Select).

Setting the field's *Options* changes the options available for selection. You're able to add and remove options as well as edit each individual option's display name and value (only available for Radio and Select fields).

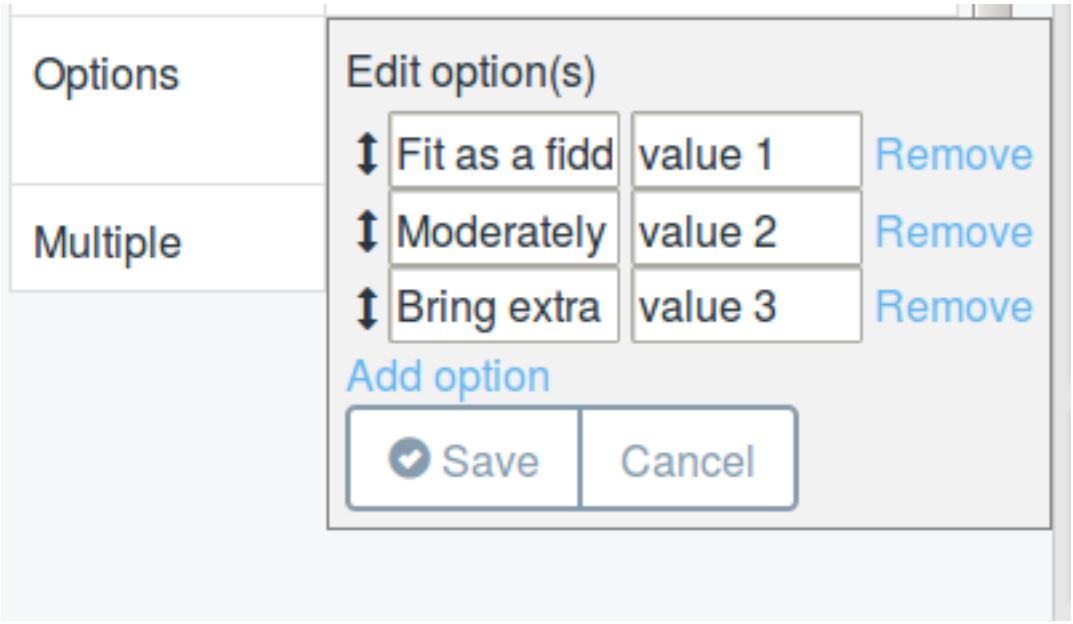


Figure 24.16: : The properties of a definition's data fields are configurable. For example, add and edit options for the *How fit are you?* Select field drop-down menu on the Activity Entry form.

In addition to dragging the fields around on the canvas, you can stack inputs within inputs by dragging a field within another field. You can organize your data into unlimited levels of hierarchy, creating a clear, logical data model. There is also a duplicate button on each field (the button with the plus symbol icon), allowing you to easily clone any field as many times as you need.

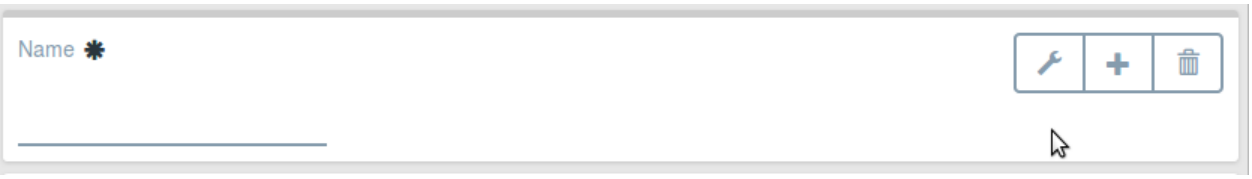


Figure 24.17: : Configure data fields by clicking the wrench icon, duplicate fields with the add button (the plus symbol icon), or delete them using the delete button (the trash can icon).

Once you have a data definition, you and your list creators can use it to start creating lists for users to fill out.

24.2 Creating Data Lists

To create list applications, it's not enough to just create data definitions, as you read about in the last article. Although the definitions are the central component to your simple list application, controlling the data that can be entered by users, you need to create a Dynamic Data List that uses the data definition. Very simply, data lists create boundaries around the records entered. For example, if you use the Activity Entry definition created in the last article to create a Lunar Rover Race Sign-Up list, the records are stored for you, even if someone else uses the same data definition to back their Lunar Hike Sign-Up list. You can display both sign-up forms on the same page, backed by the same data definition, and safely keep your records from being mixed up.

Building out new lists is similar to creating new web content articles. Just as you can create new Web Content Structures to control the input of a particular type of web content, you can use Data List Definitions to control the input of new list types. Similarly, just as you create a new piece of web content, selecting the Structure you would like to use with it, Lunar Resort site members with the proper permissions choose the Definition they want to use when creating a new list. Now that a data definition is in place, all that remains is to create a new data list to capture the information we're going after. This is the easiest step in creating the list, with only a few clicks between a data definition and robust data entry and delivery.

To create a new sign-up form and list for a lunar rover race:

1. From the Product Menu, select the Site you want to create Dynamic Data Lists in (the Lunar Resort if you're following the example), then click on *Content → Dynamic Data Lists*.

2. Click the *Add* button () to create a new list based on a data definition.

3. Give the data list a name, like *Lunar Rover Race Sign-Up*, and a description to assist administrative users in the future.

4. Last and most importantly, click *Select* under the *Data Definition* section–this is where you set the data model that drives this list.

   Choose the *Activity Entry* data definition you created, then click *Save*.

Now that you've created your brand new sign-up, get the word out for guests to sign up. But what would it look like for them to add an entry to this list? The layout of the data definition you previously created (or selected) defines the default layout of the form as well. You just need to add a Data List Display portlet to a page, and set the permissions on the list so guests can add records.

Note: If you want to allow guests to submit DDL forms, you'll need to enable the *Allow Multiple Submissions* feature. Liferay treats the Guest user as one user. Therefore, if a guest submits a DDL form, Liferay assumes the Guest user filled out their one form and it won't allow any other guests to submit a form. Thus, allowing the Guest user to submit multiple submissions allows all your guests to submit DDL forms.

But how will the list of entries appear on a page? Whether its the awesome, new Lunar Rover Race Sign-Up sheet or that boring Jobs Listing, the display side of the equation is very important.

Using Default Displays

Lists are published in the portal through the Dynamic Data List Display portlet. If list creators don't customize the display, their lists use the default display template.

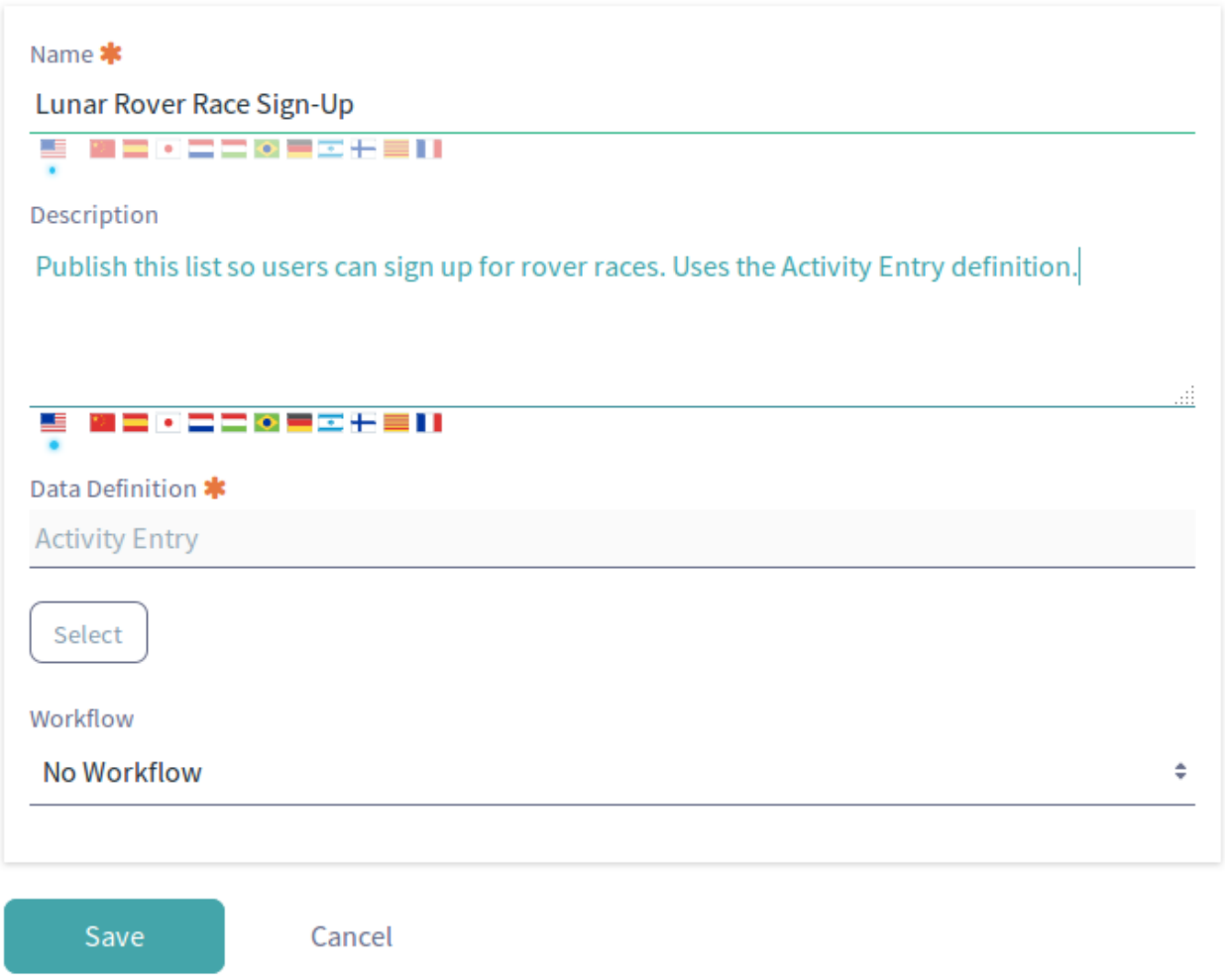


Figure 24.18: Give your list a name, a description, and select a data definition for the list.

The default display template isn't exciting, but it allows users to see the list's contents, and if they have permission, to add and/or edit list items. This type of interaction is used for display-only lists that the user chooses to expose to others, or for the user's own private lists. But you can improve the display. You can show the data in a spreadsheet, so you can view the responses to your Lunar Rover Race Sign-Up in a comfortable, easy-to-read format. The Dynamic Data List Display portlet provides an easy way for a user (such as a member of a site) to interact with whatever list is active and available.

While it's possible to ask everyone to contribute to the data list in the control panel, it's simpler if the list is placed on a public or private page in your site. Liferay's Dynamic Data List Display portlet lets you do just that. Just navigate to whichever page you want and add the portlet. It works much like the Web Content Display portlet. You can select a list for display, add a list, or edit the currently displayed list. You can also add a *Form Template* or a *Display Template* to modify the appearance of your lists. This will be covered in the next article.

Open the Dynamic Display Portlet's configuration menu by clicking the Actions button ( ) and then clicking *Configuration*. From this menu, you can select the list to display and configure it how you like. You can select a different Display or Form Template, or choose whether the list appears as *Editable* and whether to use the *Spreadsheet View*. Checking *Editable* allows users that have permission to add new entries to the list.

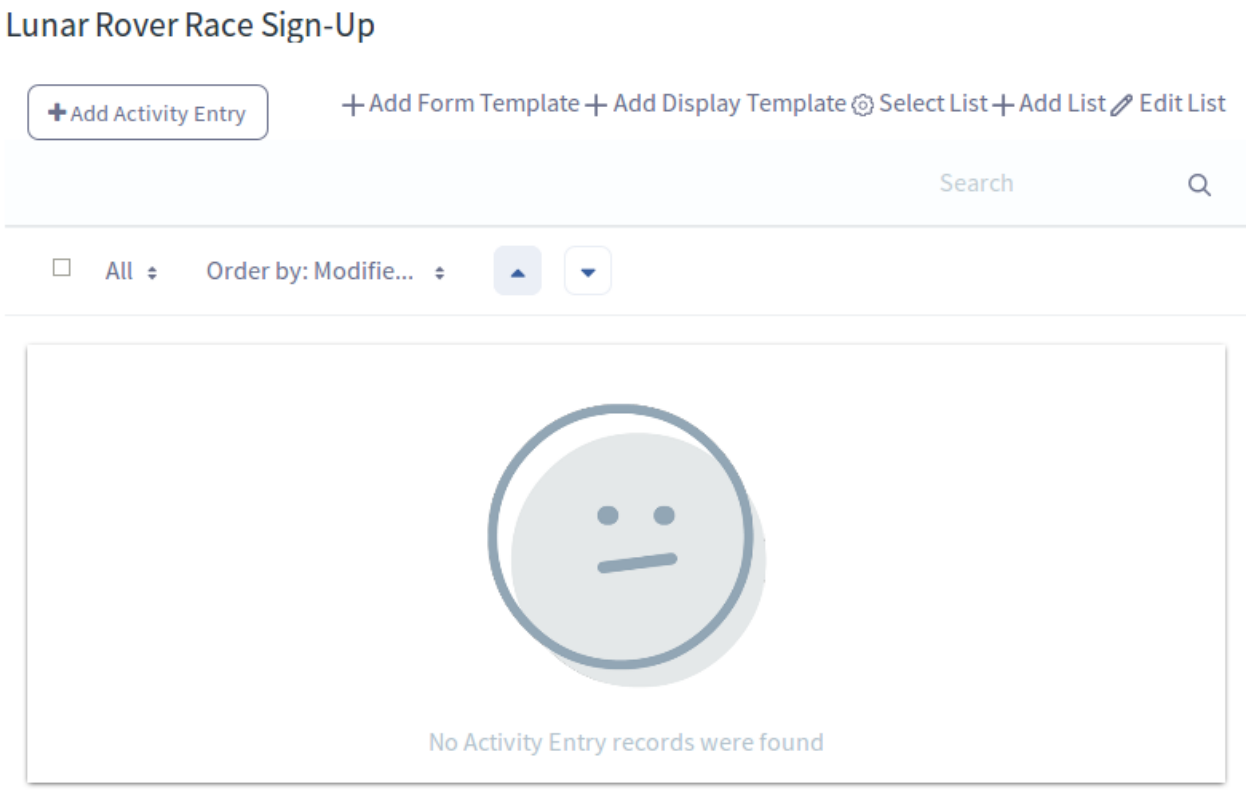


Figure 24.19: You can enter a new data record directly from the Dynamic Data Lists portlet by clicking on *Content → Dynamic data Lists* from the from the Product Menu, clicking the list you want to add a record for, then click the *Add* button. Alternatively, you can add the Dynamic Data List Display portlet to a page and configure it to allow users to submit new data records.

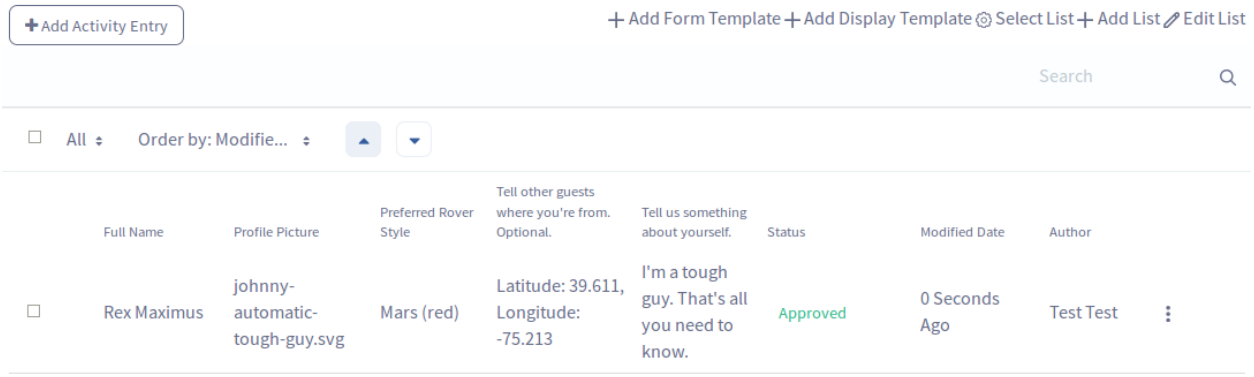


Figure 24.20: The default data list display in the Control Panel shows the list's records and allows record to added, edited, or removed (assuming the user has the correct permissions).

By default, this is enabled, but administrators are the only ones with add permission. To grant access to other users, edit the permissions on the list you'd like to grant access to, and grant the `Add Record` permission. Choosing to use the *Spreadsheet View* displays the list in a dynamic spreadsheet view. This allows users with permission to interact with the list in the same way as in a standard spreadsheet program.

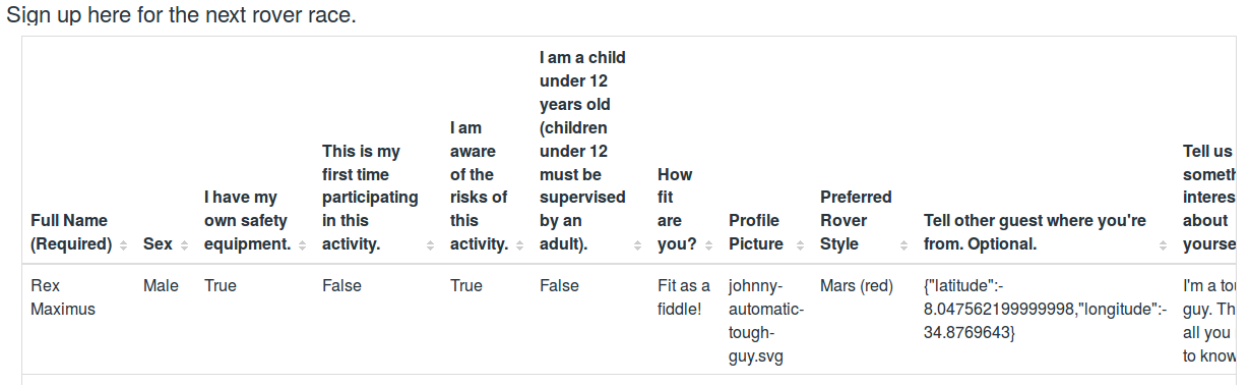


Figure 24.21: The Dynamic Data List Display portlet allows users to view the records belong to a list in either a standard or a spreadsheet view. This figure shows the spreadsheet view.

As useful as the default display is, and it's certainly useful for data lists like to do lists and memo notes, it can be an awkward way to ask Lunar Resort guests to sign up for activities. Almost any time you want users to interact with lists and contribute responses, you'll want a simple form to show them, and you won't want to display all of the data that gets stored for each list entry. Next, learn how to leverage template editors to control the display of your forms and lists.

24.3 Using Templates to Display Forms and Lists

After creating data definitions and lists, you can control how the form appears to your users, and how the resulting list of records is displayed. You do this by creating templates for each view (form view and record display view) and selecting them in the DDL Display portlet. You might realize you need to create a sign-up sheet with a subset of the fields provided by the Activity Entry data definition created in the article on data definitions last two articles. Instead of creating a new definition and list, just customize the data definition's templates. Another use case would be a situation where one of the preexisting data definitions that Liferay offers suits your list needs, but you need it laid out differently, or need to provide different labels or field configuration options. In that case, create a list from the definition, then add form and/or display templates to the data definition.

Data definitions can have as many form and display templates as you care to create (or none, if you're satisfied with the default templates). Choose whichever template you want to use for your lists in the Dynamic Data display portlet.

Managing Display Templates and Form Templates

Since Display and Form Templates correspond to a particular data definition, they're accessed from the *Data Definitions* section of the Dynamic Data Lists portlet.

- Open the Product Menu, select the Site you want to work in (the Lunar Resort), then click *Content* → *Dynamic Data Lists*.

- Open the Configuration menu by clicking the Actions button ( ) at the top right corner of the portlet, and select *Manage Data Definitions*.
- Find the Activity Entry definition, then open its Actions menu by clicking the Actions button ( ).
- Select *Manage Templates* to begin working with the definition's templates.

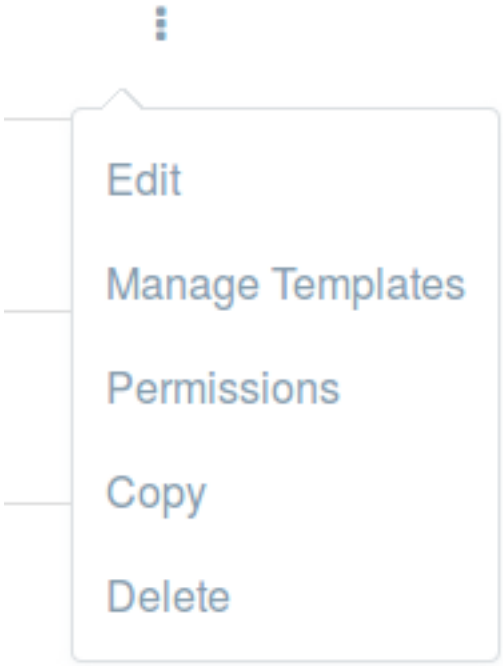


Figure 24.22: Manage and create display and form templates to change the display of forms and lists.

If there are any templates for the definition, you'll see them listed. To edit, copy, delete, or configure permissions for the definition, open its Actions menu by clicking the vertical ellipsis icon.

If no templates have been created for a definition, you can create new ones. See the next sections on Creating Form Templates and Creating Display Templates for more details.

Creating Form Templates

The default data entry form is the entire data model from a data definition, including all required and optional fields. List creators who create new lists using a data definition will see every item in that definition on the input form. But what if you just want a quick sign-up form to find out who's coming to the Lunar Luau dinner tonight? Using a form template for an existing data definition, you can customize the display of the Activity Entry definition's form. If you don't know what the Activity Entry data definition is, refer to the article on data definitions, where it was created.

You can access and create new form templates from the Product Menu:

- Click *Content* → *Dynamic Data Lists*. Make sure you're in the correct site scope (to follow the example, select the Lunar Resort site).
- Open the configuration menu by clicking the vertical ellipsis icon at the top right-hand corner of the page.

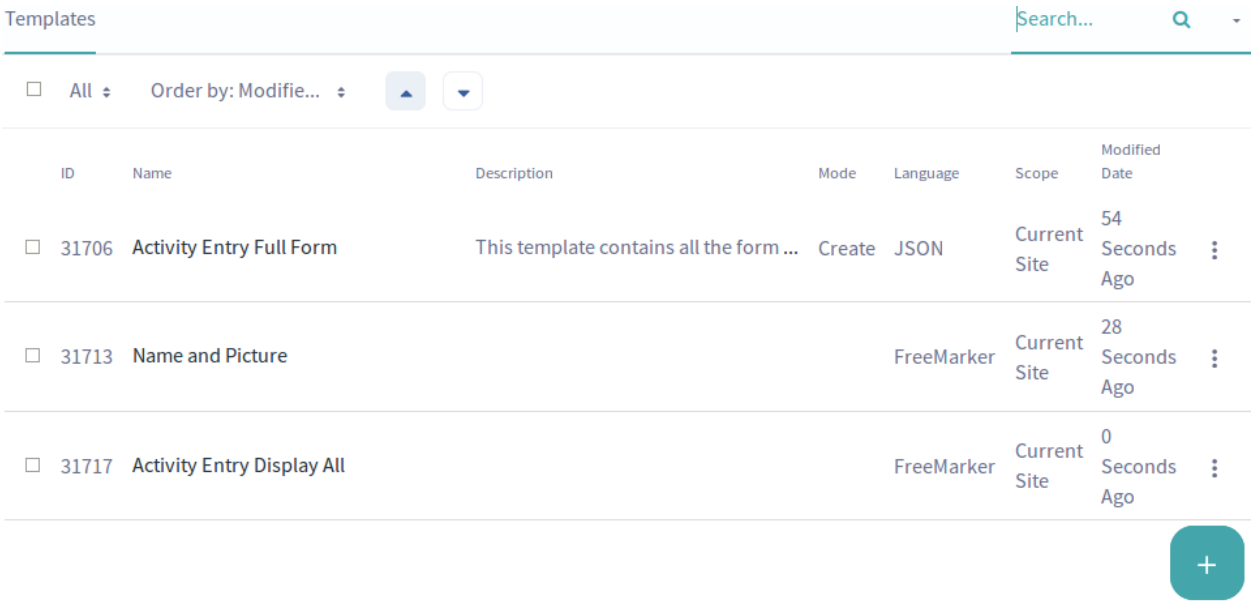


Figure 24.23: Manage all the templates for a data definition from the Manage Data Definitions section of the Dynamic Data Lists Portlet.

- Select *Manage Data Definitions*, then click on the *Actions* button next to your data definition of choice and select *Manage Templates*.
- Click the Add button, then *Add Form Template*, you're presented with the same kind of graphical, drag-and-drop interface used for creating the data definition.
- Move items around, delete unwanted fields from view and save when ready.You can limit the fields displayed for entry, change the order of elements, or provide more accurate labels.

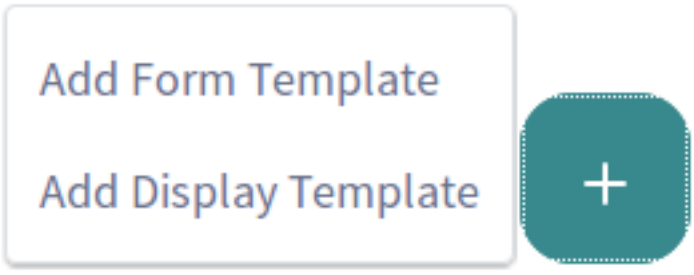


Figure 24.24: Add display and form templates to change the display of forms and lists.

Alternatively, you can create form templates from the Dynamic Data List Display Portlet:

- Navigate to the page with your DDL Display portlet and make sure the appropriate list is selected for display.
- Find the *Add Form Template* icon on the bottom-left corner of the portlet window and click it to create a new template. Make sure you're signed in as a user with permission to create templates.

Figure 24.25: Add display and form templates directly from the Dynamic Data Lists Display portlet.

With Form Templates, can use one data model that encompasses the maximum information you might need, then quickly produce form templates that make it look like you have several data definitions and lists. How will you display all the records, though? However you want, of course.

Creating Display Templates

For every data definition, you can create as many displays as you need. If you've created a form template that doesn't show all the fields of a particular data definition, you probably don't want to display those fields in the list, either. For example, if you created a special "Lunar Luau Dinner Sign-Up" list using your "Activity Entry" definition, you wouldn't want to confuse Lunar Resort guests by displaying data fields you never asked for. "Preferred Rover Style?" a guest might say. "I don't remember seeing *that* on the sign-up form!" To avoid such embarrassing situations, you should create a custom display to match that list. Take it even further and provide a fancy, JavaScript-driven image carousel preview of all the attendees of the party. Display templates give you the power to do all this and more.

The first task is to create a new display template for the Volunteer Sign-Up data definition. As with Form Templates, you can do this from the Dynamic Data List Display portlet, or from the Dynamic Data Lists portlet in the Product Menu.

From the Dynamic Data List Display portlet:

- Navigate to the page with your DDL Display portlet and make sure your list is selected for display.
- Find the *Add Display Template* icon on the bottom left corner of the portlet window and click it to create a new template. Make sure you're signed in as a user with permission to create templates.

Figure 24.26: Add display and form templates directly from the Dynamic Data Lists Display portlet.

From the Product Menu:

- Click on *Content → Dynamic Data Lists*.
- Open the configuration menu (click the vertical ellipsis in the portlet's title bar), then click *Manage Data Definitions*.
- Find your data definition in the list, then click the Actions button and choose *Manage Templates*.
- Now click the Add button and choose *Add Display Template*.

Fill out the form with a name and a description. Next, choose whether to use FreeMarker or Velocity to write your template.

Once you choose the script language, you can upload a template file or use the template editor to enter a script manually. Inside the editor, you have access to a palette featuring common variables related to your

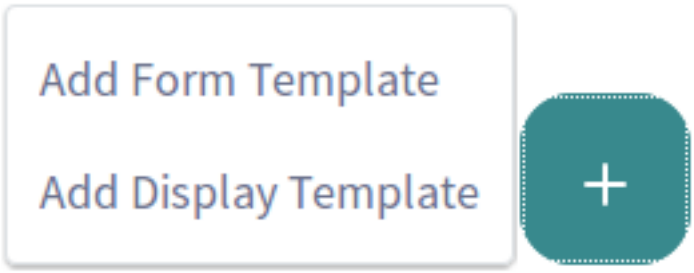


Figure 24.27: Add display and form templates to change the display of forms and lists.

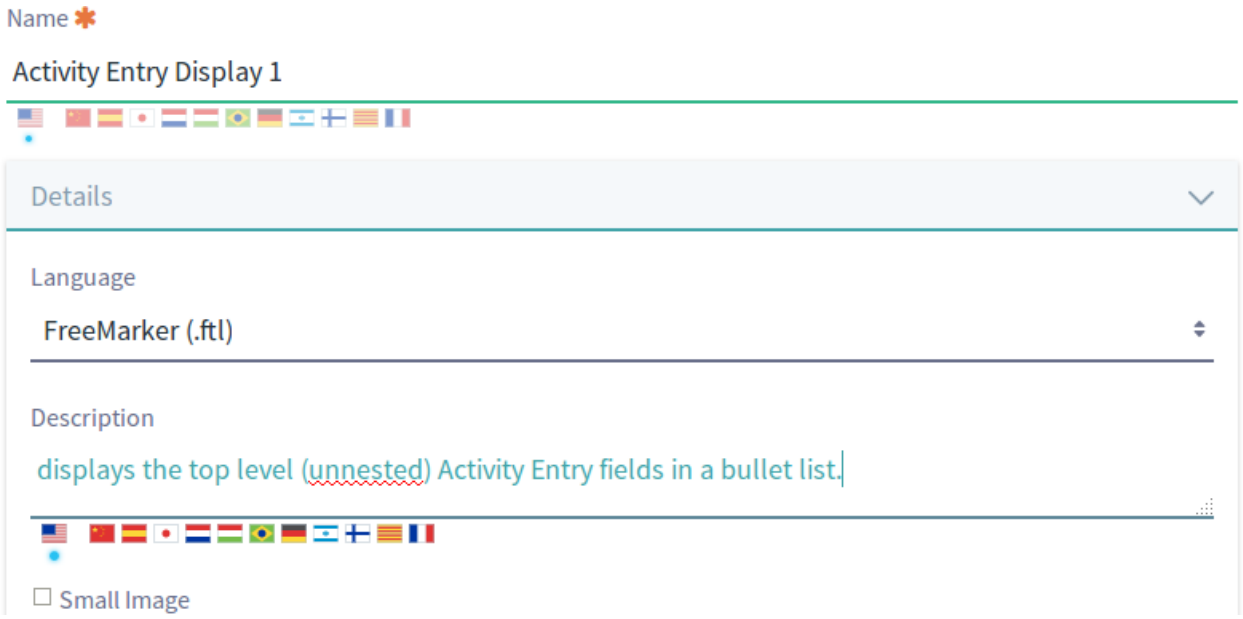


Figure 24.28: Give your Display templates a name, description, and choose whether to use FreeMarker or Velocity in your template.

selected template language. Hover your pointer over a variable in the palette for a more detailed description. To place a variable into the display template code, position your cursor where you want it placed, and click the variable name in the palette. Another useful tool in the display template editor is the autocomplete feature. In a FreeMarker template, it can be invoked by typing *${* which opens a drop-down menu of common variables. Upon selecting one of the variables, the editor inserts the variable into your display template code.

Note: If you're familiar with Web Content Templates, Display templates customize the display of a list in the same way. Display templates can be written in FreeMarker or Velocity, pulling data from the data definition in the same way that web content templates pull data from their structures. Also similar to web content templates, display templates can be embedded in other display templates. This allows for reusable code, JS library imports, or macros which will be imported by Velocity or FreeMarker templates in the system. Embedding display templates provides a more efficient process when you have a multitude of similar data definitions. Just import an embedded display template and work off of it for your new display template. We'll look at a simple example, but for more information on using template scripts to pull data from a backing structure, see the article on web content templates.

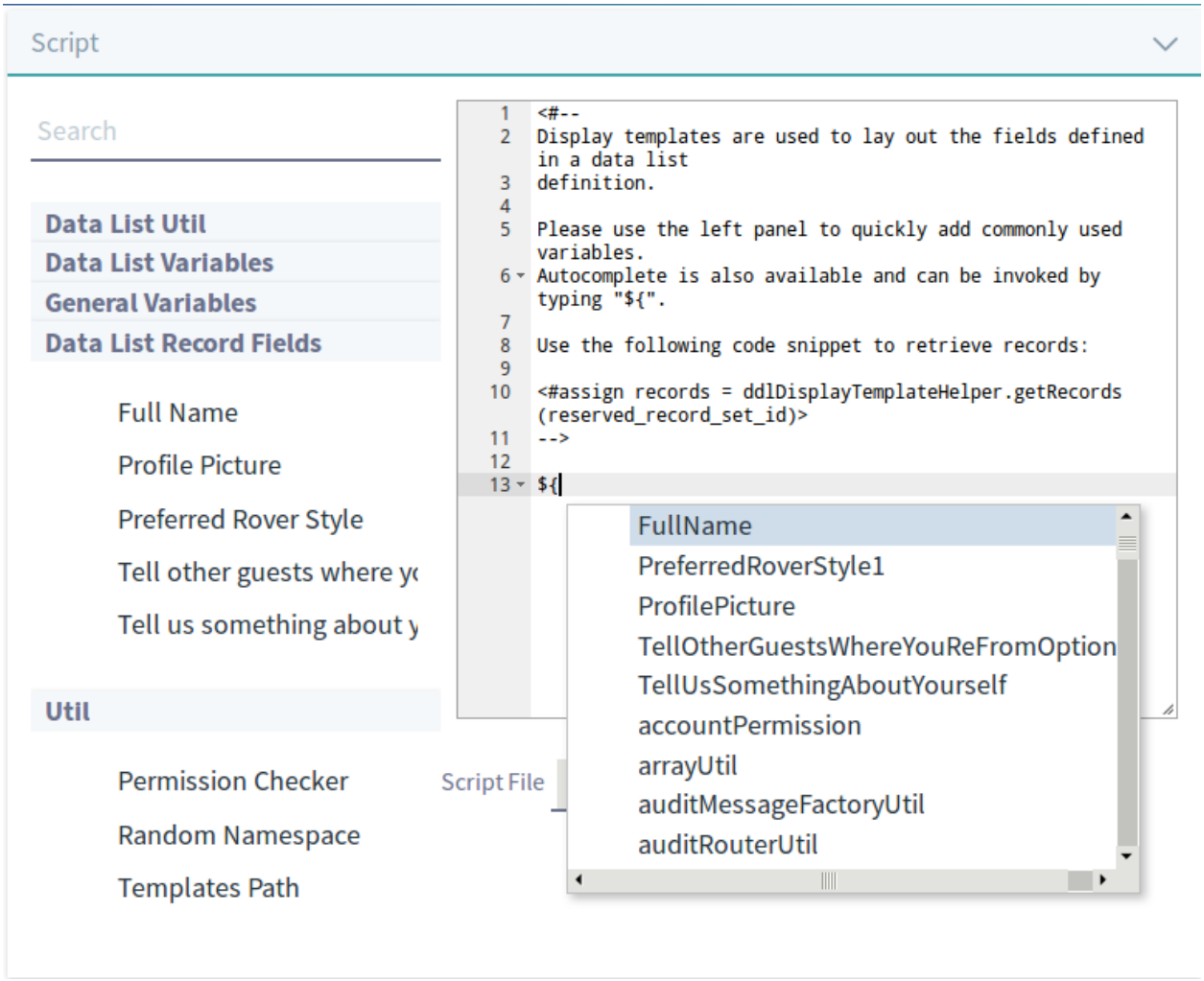


Figure 24.29: Upload an existing template, or create one using the template editor.

Creating a Simple Display Template There's a lot you can do with your templates. Maybe there's a Lunar Luau dinner scheduled at the Lunar Resort, and the kitchen staff wants a count of the guests. You could create a list based on the Activity Entry data definition you created earlier, make a form template that just shows the *Full Name* field and the *Tell us something interesting about yourself.* field. Then write a FreeMarker template that gives a summary of who is attending, displaying just the fields form the form template. You jsut need to access the appropriate records for the list and display them. Use the available helper variables to find out what records you have access to:

```
reserved_ddm_template_id

reserved_ddm_structure_id

reserved_record_set_description

reserved_record_set_id

reserved_record_set_name
```

Inside a template, these variables give us the ID for the record set (that contains all of the volunteers in

our list), as well as the name, description and data definition. You can easily retrieve all the data list's records using the handy `DDLDisplayTemplateHelper` class, which provides the ability to perform some common tasks without any difficulty. These are the functions available using `DDLDisplayTempalteHelper`:

```
getDocumentLibraryPreviewURL

getHTMLContent

getLayoutFriendlyURL

getRecords

renderRecordFieldValue
```

For example, if you want to use the `getRecords` method to have access to all of a data definitions existing entries, enter

```
<#assign records = ddlDisplayTemplateHelper.getRecords(reserved_record_set_id)>
```

into your Display Template. This *fetches* the records of the data list (really,the list's data definition) definition the template is associated with. You haven't done anything with them yet, so your display is still empty. To list all the records, you need to use the *Data List Records* helper on the right hand column of the template editor. Remember to place your cursor in the proper place in the template editor window, then click *Data List Records*. You'll see

```
<#if records?has_content>
    <#list records as cur_record>
        ${cur_record}
    </#list>
</#if>
```

appear wherever you placed your cursor. You might think you'll have a nicely formatted list of all the data records, but you won't. It will spit out everything in the database for the given data definition, which is ugly and practically useless.

{uuid=139bc32a-398a-d3c3-aa74-bfa0476f7093, recordId=20923, groupId=20146, companyId=20118, userId=20163, userName=Test Test, versionUserId=20163, versionUserName=Test Test, createDate=Wed Oct 14 17:02:37 GMT 2015, modifiedDate=Wed Oct 21 16:12:20 GMT 2015, DDMStorageId=22135, recordSetId=20902, version=1.2, displayIndex=0, lastPublishDate=null} {uuid=68a0dff4-be1d-2110-99af-5350fa112592, recordId=22214, groupId=20146, companyId=20118, userId=20163, userName=Test Test, versionUserId=20163, versionUserName=Test Test, createDate=Wed Oct 21 16:41:07 GMT 2015, modifiedDate=Wed Oct 21 16:41:07 GMT 2015, DDMStorageId=22215, recordSetId=20902, version=1.0, displayIndex=0, lastPublishDate=null} {uuid=a97c79ca-54b3-18c4-84c4-2289dc0cf2b7, recordId=22233, groupId=20146, companyId=20118, userId=20163, userName=Test Test, versionUserId=20163, versionUserName=Test Test, createDate=Wed Oct 21 16:47:37 GMT 2015, modifiedDate=Wed Oct 21 19:24:32 GMT 2015, DDMStorageId=22460, recordSetId=20902, version=1.2, displayIndex=0, lastPublishDate=null} {uuid=a7516119-6f6b-6379-b021-ed2527bd9ebe, recordId=22424, groupId=20146, companyId=20118, userId=20163, userName=Test Test, versionUserId=20163, versionUserName=Test Test, createDate=Wed Oct 21 18:40:05 GMT 2015, modifiedDate=Wed Oct 21 18:57:26 GMT 2015, DDMStorageId=22440, recordSetId=20902, version=1.1, displayIndex=0, lastPublishDate=null}

Figure 24.30: It's not useful to simply get the data records and display them in their raw form.

You're getting closer to having a coherent display template. There's just a couple steps left. First, wrap the line with `${cur_record}` in HTML tags that create an unordered list, with the records as list items:

```
<ul><li>
${cur_record}
</li></ul>
```

If you look at the list now, it's getting more organized. Each record is set apart with a bullet point, and it's a bit more readable. But you don't need all the confusing database fields displayed to your users. Narrow it down to simply the *Full Name* field and the *Tell us something interesting about yourself.* field. Right now you're listing all the fields for each record by specifying `${cur_record}`. If you remove that, and instead use the template editor's *Data List Record Fields* to explicitly list which fields should be displayed, you can easily finish the display template. For example, if you want to put the Name field in, place your cursor inside the `<li>` tag and click the *Name* entry under *Data List Record Fields*. You'll see that the FreeMarker code

```
${ddlDisplayTemplateHelper.renderRecordFieldValue(cur_record.getDDMFormFieldValues("FullName")?first, locale)}
```

was added. That's nice, but if you want the format for each bullet to read
[Full Name] is coming and wants you to know this:[Tell us something interesting...]
you need to replace the name field you added with the following lines:

```
<em>${ddlDisplayTemplateHelper.renderRecordFieldValue
        (cur_record.getDDMFormFieldValues("FullName")?first, locale)}</em>
            is coming and wants you to know this:
${ddlDisplayTemplateHelper.renderRecordFieldValue(cur_record.getDDMFormFieldValues("TextBoxadfn")?first, locale)}
```

There's the meat of the template. Wrap the `FullName` field in an `<em>` to emphasize it, add the text "is coming and wants you to know this:" and then get the value of the `TextBoxadfn` field (that's the field name for your *Tell us something interesting...* field for each record.

To complete the template, add a heading and a simple description of the list before the list items:

```
<h1>Guest List for ${reserved_record_set_name}</h1>

These people are attending the Lunar Luau! You should too! <br />
```

Now save the display template. If you've added a list to a DDL display portlet already, configure it to use the new Display Template, add a record, and see how it looks.

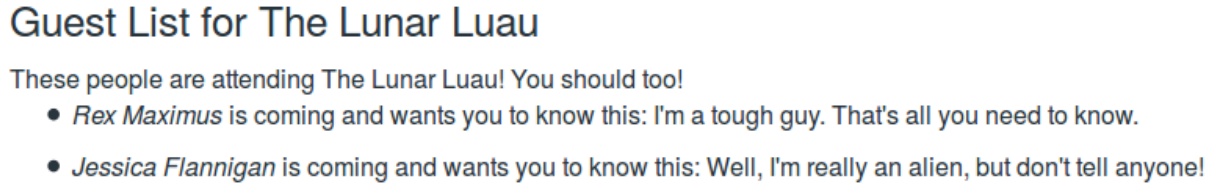


Figure 24.31: : This display template provides a list of guests attending the event and an interesting fact about each guest.

That gives you a simple, clean display of only the form data that you want your site guests to see. Here's the whole template:

```
<#assign records = ddlDisplayTemplateHelper.getRecords(reserved_record_set_id)>

<h1>Guest List for ${reserved_record_set_name}</h1>

These people are attending ${reserved_record_set_name}! You should too! <br />

<#if records?has_content>
    <#list records as cur_record>
            <ul><li>
                <em>${ddlDisplayTemplateHelper.renderRecordFieldValue
                    (cur_record.getDDMFormFieldValues("FullName")?first, locale)}</em>
```

```
                    is coming and wants you to know this:
                ${ddlDisplayTemplateHelper.renderRecordFieldValue(cur_record.getDDMFormFieldValues("TextBoxadfn")?first, locale)}
            </li></ul>
    </#list>
</#if>
```

Since that was so simple and fun, what if you wanted to display the *Full Name*, *Interesting fact...*, and *Profile Picture*?

```
<#assign records = ddlDisplayTemplateHelper.getRecords(reserved_record_set_id)>

<h1>Here are the people currently signed up! Join them now! Click the Add button above to fill out the form!</h1>

<#if records?has_content>
    <#list records as cur_record>
        <ul>
            <li>
                <#-- The below FreeMarker code gets the Preview URL of the Profile Picture and displays it using an HTML img tag -->
                <img src="${ddlDisplayTemplateHelper.getDocumentLibraryPreviewURL(cur_record.getDDMFormFieldValues("ProfilePicture")?first, locale)}" height

                <#-- The below FreeMarker code gets the Interesting Fact field and wraps it in an <em> tag -->
                About me: <em>${ddlDisplayTemplateHelper.renderRecordFieldValue(cur_record.getDDMFormFieldValues("TextBoxadfn")?first, locale)}</em><br />

                <#-- The below FreeMarker code gets the Full Name field -->
                ${ddlDisplayTemplateHelper.renderRecordFieldValue(cur_record.getDDMFormFieldValues("FullName")?first, locale)}
            </li>
        </ul>
    </#list>
</#if>
```

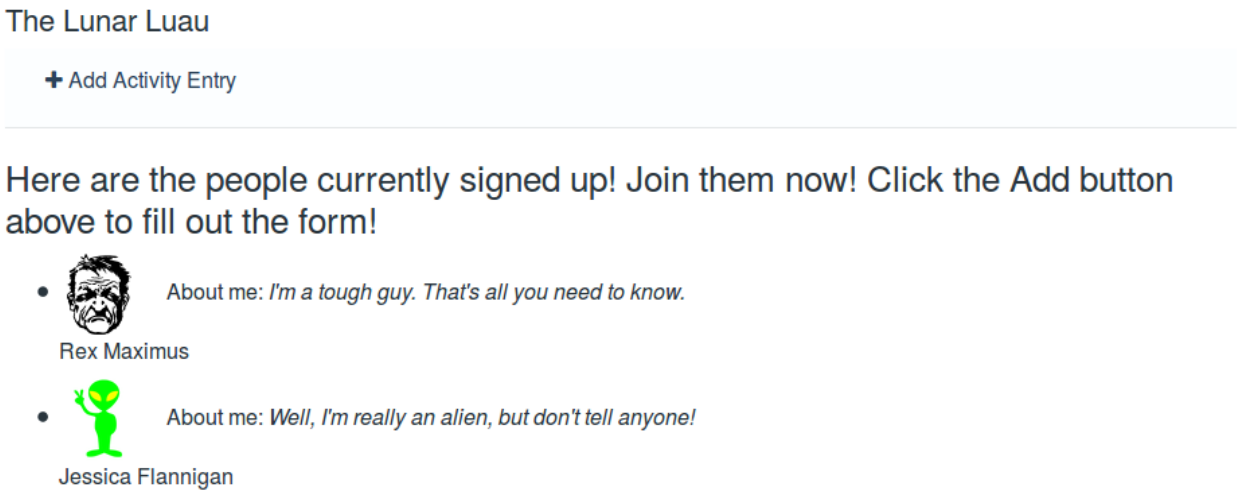


Figure 24.32: : This display template shows the *Full Name* field, a small version of the image uploaded as a *Profile Picture*, and the *Tell us an interesting...* field.

As you can see, using the template editor's functionality to access a Data Definition's records and fields is easy, and you can assemble a display template quickly. Add some HTML to improve the formatting, and you now have an attractive display to present your data.

With the full power of FreeMarker or Velocity templates at your fingertips, you have easy access to all the data of your lists, as well as the full complement of helper methods and the Alloy UI JavaScript library to make easy work of dynamic displays.

Over the last few articles, you've gained enough knowledge to begin creating your own simple applications. Using Liferay's Dynamic Data Lists portlet, the Dynamic Data List Display portlet, and the flexible power of data structures and templates, you can create list-based applications in Liferay very quickly.

Chapter 25

Using Workflow

The ancient Greeks were pretty important, so we decided to name Liferay's workflow engine *Kaleo*. In Greek, Kaleo means "called ones," which is appropriate for a workflow engine that calls users to participate in a review process designed for them. Kaleo workflow allows a user to define any number of simple to complex business processes/workflows, deploy them, and manage them through a portal interface. The processes have knowledge of users, groups and roles. You don't have to write a single line of code to accomplish this: all you have to do is create one XML document.

There are several aspects to workflow that you need to understand:

- Enabling workflow
- Designing review processes
- Uploading workflow definitions
- Configuring assets to use workflows
- Using workflow with other applications

After all that, you should be familiar with how to use Liferay's Kaleo workflow to set up approval process for any kind of content before it is published to your portal.

25.1 Embedded Workflows

In addition to the Single Approver definition, there are some workflow definitions that ship with Liferay but are not pre-installed, since they're primarily included for test cases. They can be found in the Liferay source code in `modules/aps/forms-and-workflow/portal-workflow/portal-workflow-kaleo-runtime-impl/src/main/resources/META-INF/definitions`, or in your Liferay installation itself. Open your Liferay installation's `osgi/marketplace/Liferay CE Forms and Workflow.lpkg`, and then find and open the `com.liferay.workflow.kaleo.runtime.impl-[version].jar`. The definitions are in the `META-INF/definitions` folder (see the diagrams here). Once you have the XML definition files locally, upload them into Liferay:

1. Navigate to Control Panel → Configuration → Workflow Definition.
2. Click the Add button ( ).

3. Give the workflow a descriptive title, then click *Select File* and find the XML file. Once uploaded it appears in the list of definitions.

Alternatively, you can look at the pre-existing definitions to get ideas while designing and building your own definition.

25.2 Workflow Assignees Have Implicit Asset Permissions

Users with permission to execute a workflow task (e.g., Users with the Portal Content Reviewer Role) have full resource action permissions over the assets they can review. These permissions apply in the My Workflow Tasks widget in the User's personal page and anywhere else actions on the Asset can be performed.

For example, consider a User with two permissions:

- The Portal Content Reviewer Role enables Users to review workflow submissions and grants edit and delete permissions on the content they're reviewing.
- Users also have permission to view Web Content Articles in the Site's *Content* section.

Neither permission explicitly grants the User management permissions on Web Content Articles. Users cannot normally edit or delete a Web Content Article, for example. However, if a Web Content Article is sent to the workflow, Users can access the Web Content Article for review (in their *Assigned to Me* or *Assigned to my Roles* section of My Workflow Tasks), and they can edit or delete the content while reviewing it in the workflow. While it's in the status *Pending*, they can also edit or delete the article from Site Administration → Content → Web Content because of their implicit permissions granted by the workflow system. This additional permission is temporary, and the normal resource permissions are activated once the Web Content Article exits the workflow process (for example, it's rejected or approved).

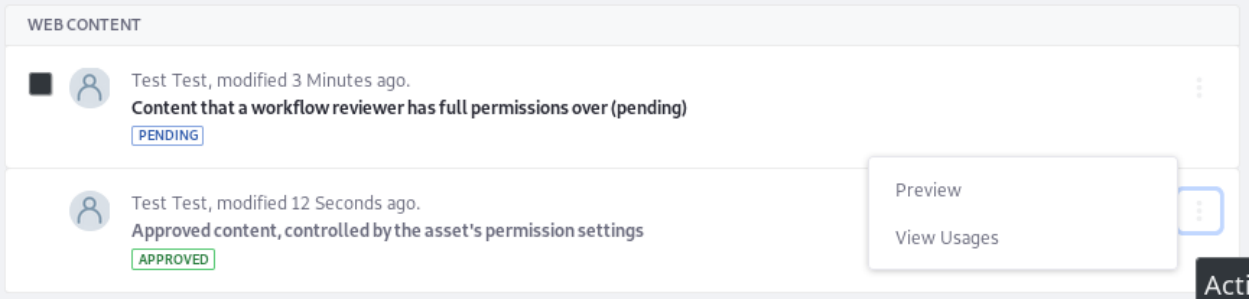


Figure 25.1: A User with VIEW permission on Web Content cannot manage Approved Articles.

25.3 Enabling Workflow

The workflow functionality bundled with the Liferay Forms and Workflow suite comes with one pre-installed workflow definition called the Single Approver Workflow. This workflow requires one approval before an asset can be published. One of the conveniences of using Liferay's workflow engine is that any roles specified in the workflow definition are created automatically when the definition is deployed. This provides a level of integration with the portal that third party engines cannot match. The Single Approver Workflow contains three roles, each with different scopes. You can deduce the scope of each role by its name: Site Content Reviewer, Organization Content Reviewer and Portal Content Reviewer.

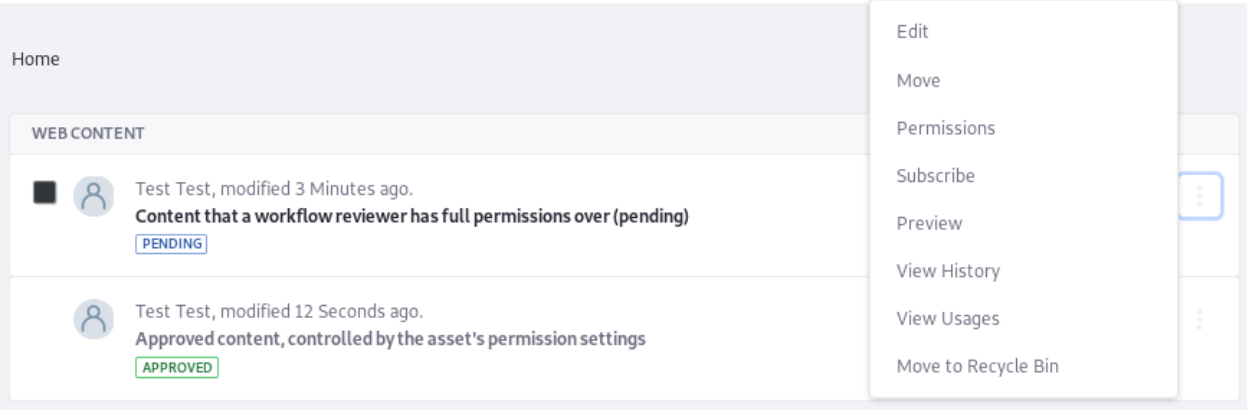


Figure 25.2: A User with access to Web Content in the Workflow can manage Pending Articles.

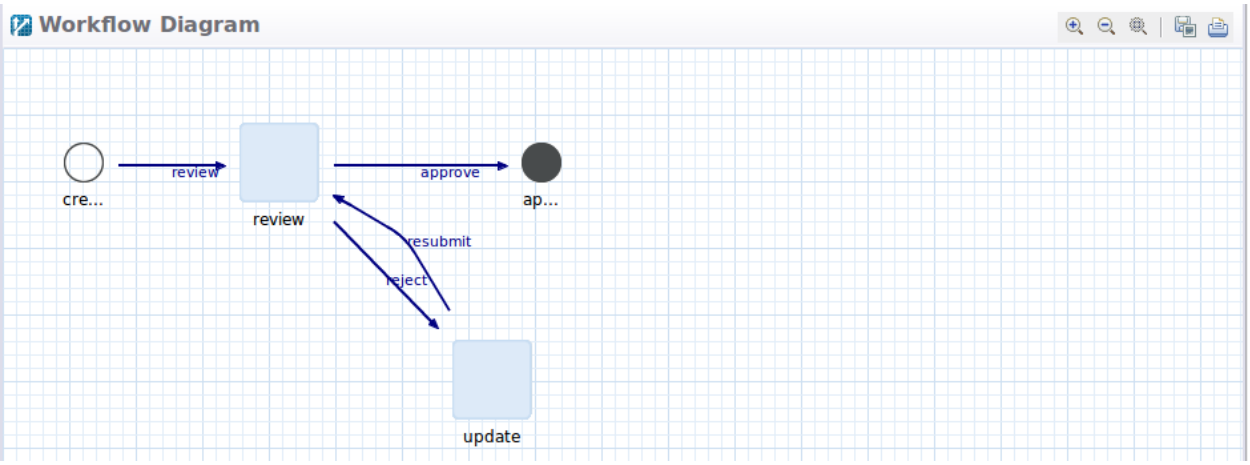


Figure 25.3: The Single Approver Definition comes included with the Liferay Forms and Workflow suite.

A word about designing workflows. If you are a CE customer, you'll have to get comfortable working with XML to create workflow definitions. There's a tutorial in the documentation for developers that will help you learn how to do that.

Workflow Assets

Many of the assets in Liferay DXP are enabled for workflow. Configure workflow or these assets in Control Panel → Configuration → Workflow Configuration:

- Blogs Entry
- Calendar Event
- Comments
- Message Boards Message
- Page Revision
- User
- Wiki Page

Configure workflow for these assets at the site level in Site → Configuration → Workflow Configuration:

- Blogs Entry
- Calendar Event
- Comments
- Message Boards Message
- Page Revision
- Wiki Page

What's the difference? As with most scoped activities, the higher level setting (in the Control Panel) sets the default behavior. It's overridden at the more granular level (in the site menu).

User doesn't appear on the site list because adding users is strictly a portal-wide administrator activity. Only assets that can be added and configured at the site level (for example, those that are accessed from the site menu) have workflow configuration controls at the site level.

Some assets that are workflow enabled are only configured in their respective application. Web Content articles are workflow enabled from the folder settings menu:

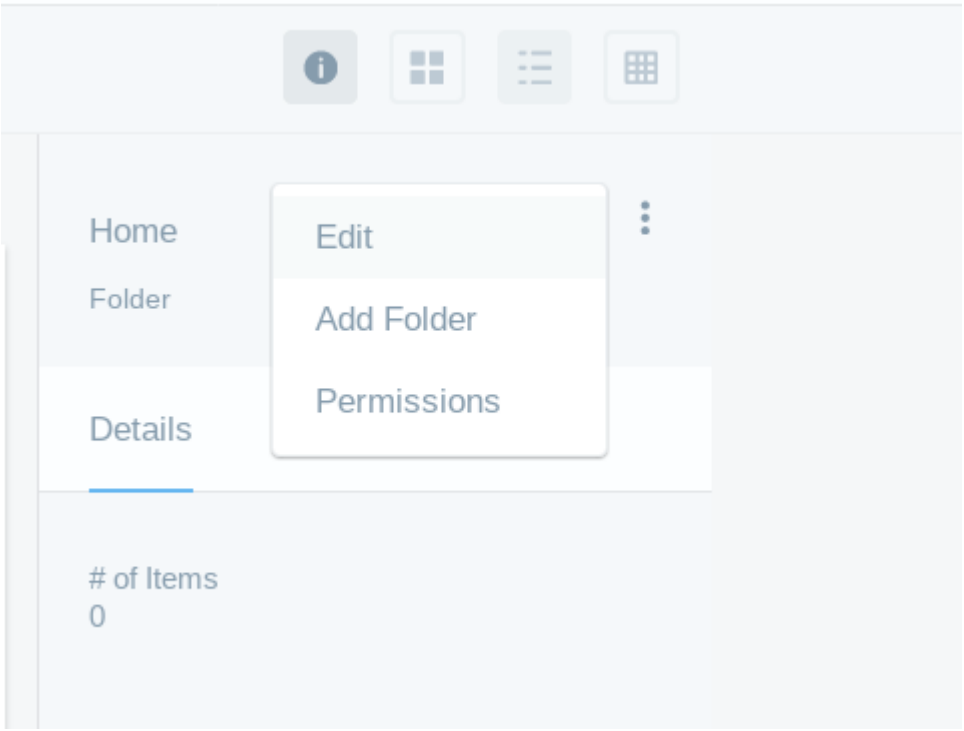


Figure 25.4: Enable workflow on Web Content at the folder level.

Enable workflow on Dynamic Data List entries in each list's Add form:

Enable workflow for each individual form's entries from the Form Settings screen:

Workflow Behavior

Most of the resources listed above behave just as you might expect with workflow enabled: The Publish button for the resource's *Add* form is replaced by a *Submit for Publication* button, and instead of instant publication, the asset has its status set as *Pending* and must proceed through the workflow before publication.

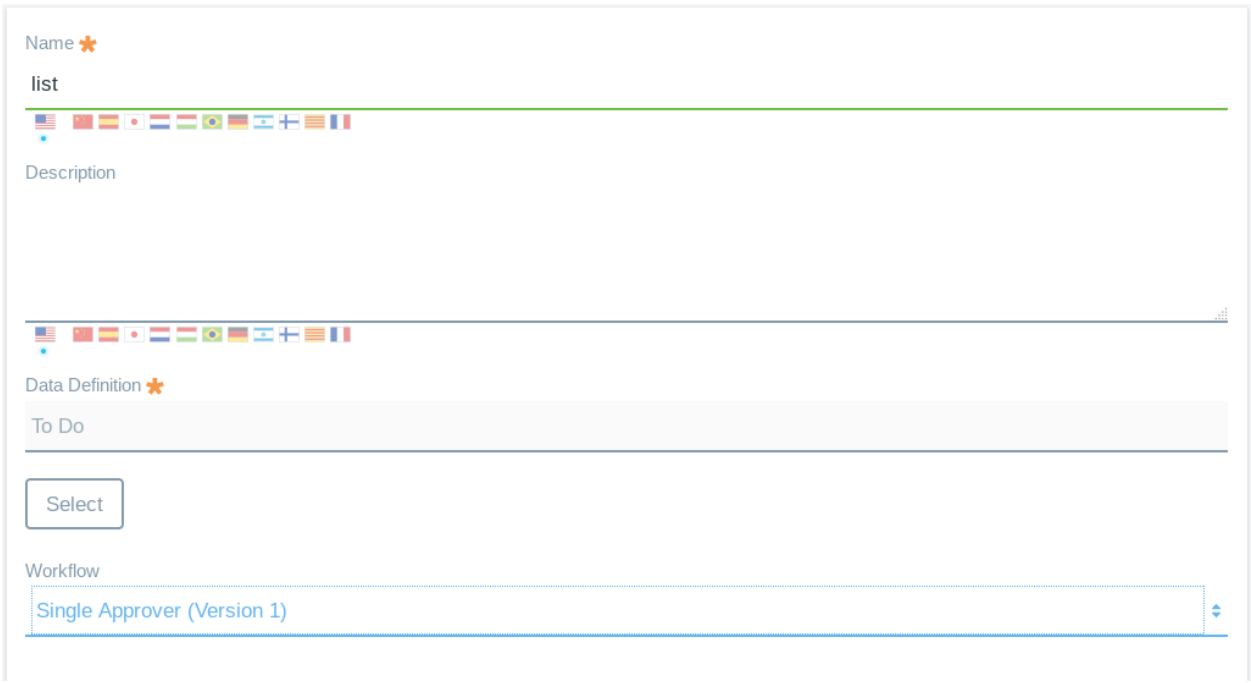


Figure 25.5: Workflow is configured for each individual Dynamic Data List.

Page revisions are slightly different. Page revisions only occur in staging environments that have Page Versioning enabled. When a Page Variation or Site Page Variation is created, its creator must click *Submit for Publication* at the top of the page, and the variation must be approved in the workflow, before it can be published to the live site.

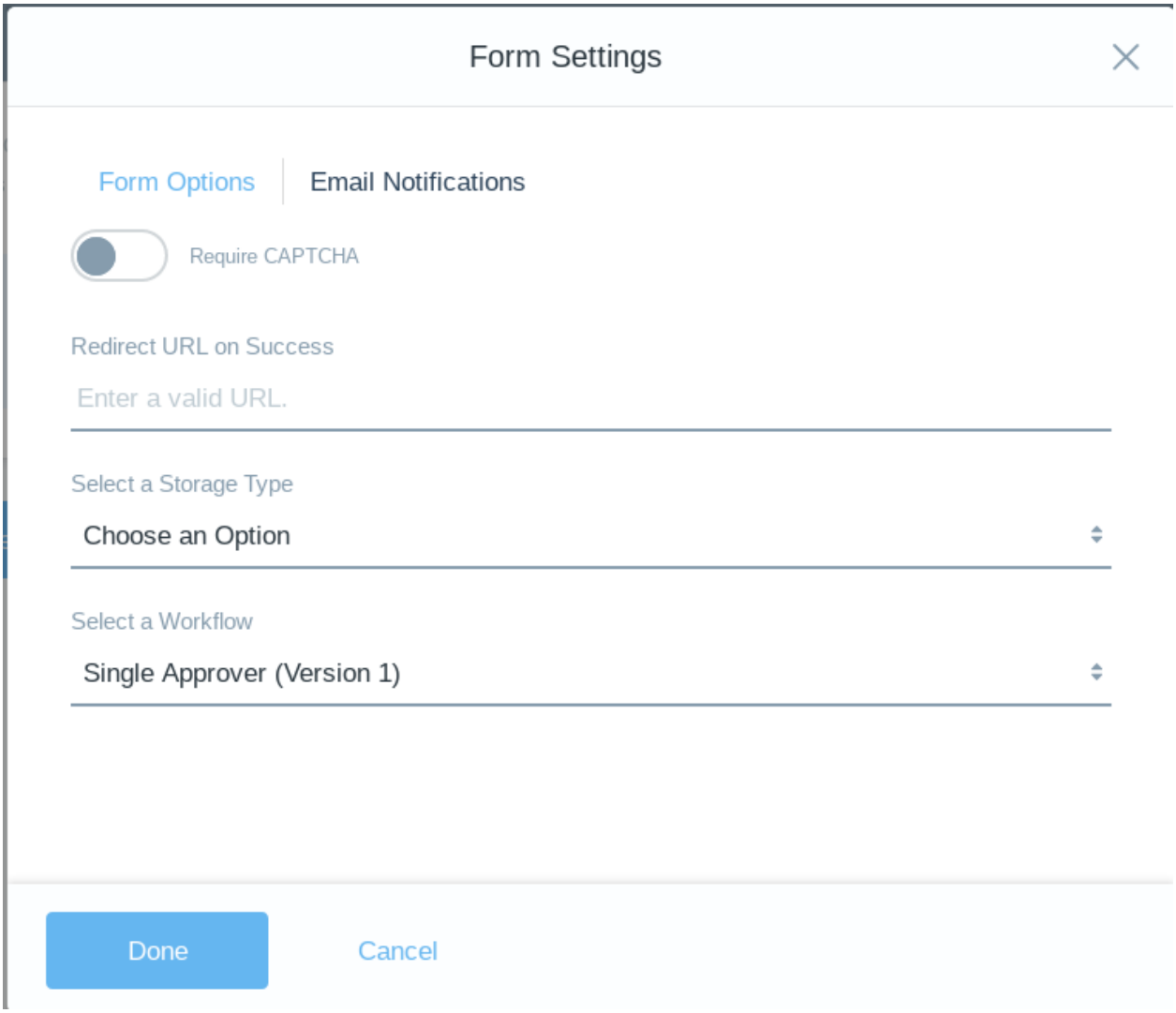


Figure 25.6: Enable workflow on each form's entries in the Form Settings window.

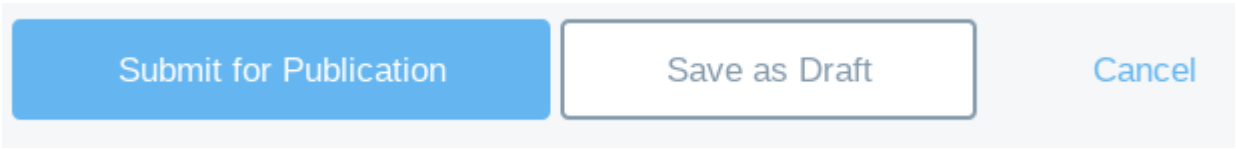


Figure 25.7: Instead of a Publish button, a Submit for Publication button appears for workflow-enabled resources.

Figure 25.8: With workflow enabled on Page Revisions, the site administrator must submit their page variation for publication before it can go live.

Chapter 26

Kaleo Designer

With the proper permissions, users can publish assets. Even if your enterprise has the greatest employees in the world, many of the items they'll wish to publish must still be reviewed, for a variety of reasons. The Kaleo Designer in Liferay DXP lets you design workflow definitions so your assets go through a review process before publication.

Kaleo Designer lets you develop workflow definitions using a convenient drag and drop user interface, so you don't need to be familiar with writing XML definitions by hand. However, some of the features can be enhanced if you're familiar with Groovy, a Java-based scripting language Liferay DXP supports in its scripting engine. All that is to say, don't be scared off when you come to a block of code in these articles. Just decide if you need the feature and find someone familiar with Java or Groovy to help you out.

Note: By default, there's only one workflow definition that's installed in Liferay DXP: the Single Approver Workflow definition. What you might not know is that you have access to several others too. If you have a local Liferay DXP installation, look in `[Liferay_Home]/osgi/marketplace/Liferay Forms and Workflow.lpkg` and find the `com.liferay.portal.workflow.kaleo.runtime.impl-[version].jar`. Open it up and then look in `META-INF/definitions`. You'll see the following workflow definitions:

```
category-specific-definition.xml
legal-marketing-definition.xml
single-approver-definition.xml
single-approver-definition-scripted-assignment.xml
```

To work with any of these definitions in Kaleo Designer, extract them from the JAR file first. Once you have the XML files locally, add a new workflow (see the below instructions) in the designer, then open the *Source (Kaleo XML)* tab. Replace the entire contents with the contents of the XML definition file. Name the definition appropriately, and click either *Save as Draft* or Publish* (see below for more information on saving and publishing). Now you can begin exploring or modifying the definition as you'd like.

It's time to start exploring the Kaleo Designer and its features.

26.1 Managing Workflows with Kaleo Designer

The Kaleo Designer gives you a convenient and intuitive interface for creating workflow definitions, from the simplest approval processes to the most complex business processes you can imagine. It features a drag

and drop interface, workflow definition versioning, and the ability to configure definitions and their nodes. With Kaleo Designer you might never need to look at a single line of XML. If you didn't have Kaleo Designer, your entire worklfow definition would need to be hand crafted in XML. Of course, the Kaleo Designer also provides the ability to directly manipulate the XML (using the *Source* tab), if you find it convenient.

So how do you add a workflow definition?

Adding New Workflow Definitions with Kaleo Designer

Access Kaleo Designer by going to the Control Panel, then clicking Configuration → Kaleo Designer. Click the Add icon ( ).

Give the workflow definition a title and you're ready to start designing your workflow.

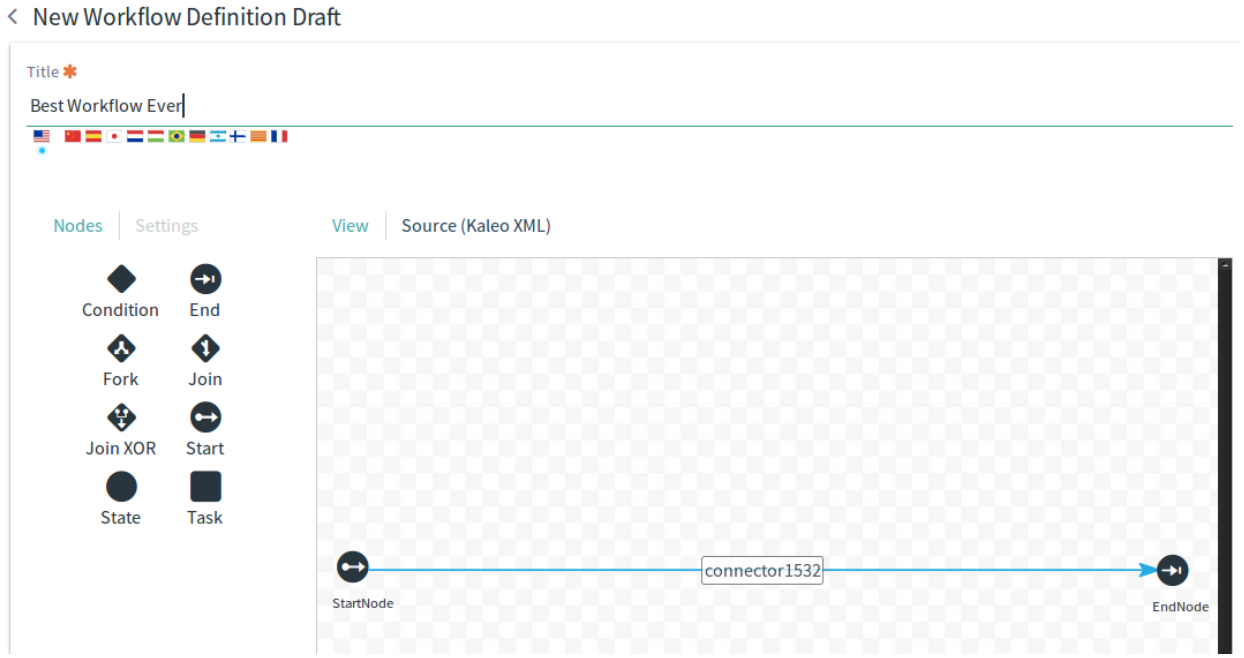


Figure 26.1: The Kaleo Designer's graphical interface makes designing workflows intuitive.

Saving and Publishing Workflow Definitions

First, look below the canvas to see the buttons that let you *Save as Draft*, *Publish*, or *Cancel*. Saving the definition as a draft lets you save your work so it's not lost (due to a timeout, for example). It won't be published (and usable in Liferay DXP), and it won't be considered a version until the Publish button is clicked. Each time you save the workflow as a draft, the Draft Version field is incremented, and each time you publish the workflow, a new Version is added.

Adding Nodes

A new workflow is already populated with a start node, an end node, and a transition between them. To make the workflow the way you want it, you need to add nodes to the workflow.

1. *Drag* a node from the *Nodes* palette on the right of the designer and *drop* it on the canvas.

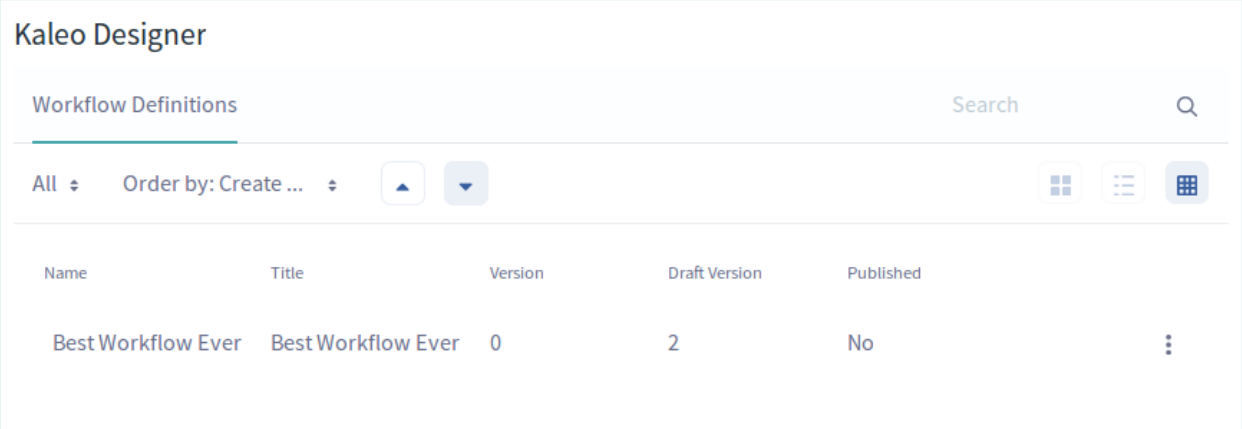


Figure 26.2: View a list of the current workflows that can be edited in the Kaleo Designer.

2. You'll see it's not connected to other nodes by a transition, so right now it can't be used in the workflow. Delete the existing transition and then you can make new transitions to direct the *flow* of your workflow (see more about transitions below if you're not sure what they're for or how to use them in Kaleo Designer).

Alternatively, start by deleting the default transition, then click the edge of the start node, drag a new transition from the start node to a blank spot on the canvas, and release it. You'll be prompted to create a node at that spot, because you can't have a transition without a starting point and an ending point on a node.

That's it. Of course, if you drag, say, a *Task* node onto the canvas, you'll want to configure it.

Node Settings

Now you know how to add nodes to the workflow definition. By default you have three things added to your canvas: a start node, a transition, and an end node. Think of the *EndNode* as the point in the workflow where an asset reaches the *Approved* status. The *StartNode* is the point at which the asset goes from the *Draft* status to *Pending*. You might decide to name your nodes to reflect what's going on in each one. To name a node, just double click it, and its *Settings* are displayed. Then double click the value of the *Name* property and you can edit the name. Click *Save* when you're done.

Of course, there's more you can do besides changing node names. Actions, Notifications, and Assignments can be used to make your workflow definition really useful and interactive. Keep reading to learn about these features.

Related Topics

Workflow Forms

Using Workflow

Liferay's Workflow Framework

Creating Simple Applications

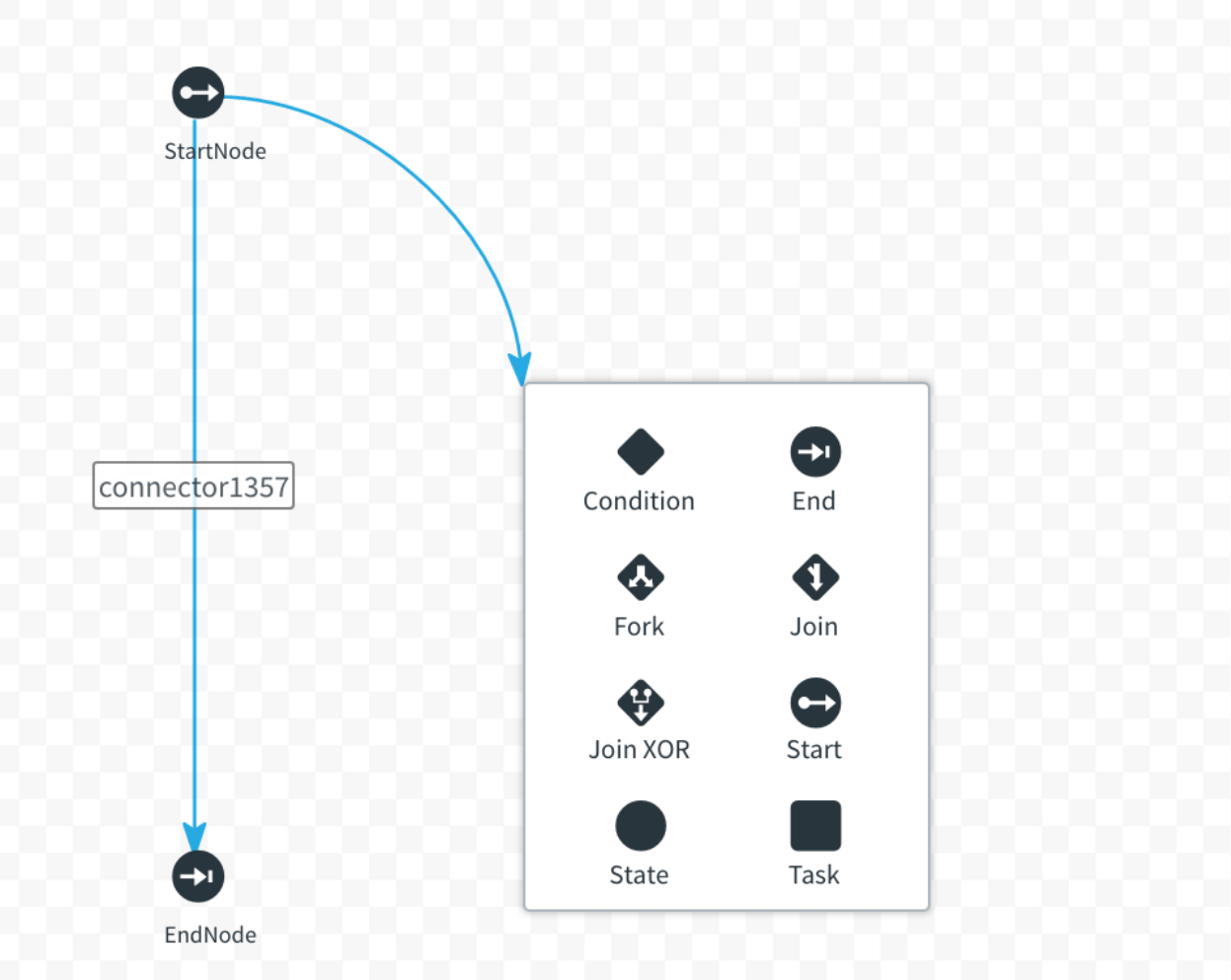


Figure 26.3: You can add a node by creating a transition that ends at a blank spot on your Designer canvas.

26.2 Workflow Definition Nodes

Once you know the basics of creating workflow definitions with the Kaleo Designer, it's time to starts getting into the details. In this article you'll learn about Actions and Notifications, two important features your workflow nodes can use. You'll also learn how to affect the processing of the workflow using Transitions, Forks, Joins, and Conditions.

There are several node types you can use in workflow definitions:

- Task nodes
- Fork and Join nodes
- Condition nodes
- Start nodes
- End nodes
- State nodes

Because they're the most complex node, and often the meat of your workflow definitions, Task nodes are covered in a separate article.

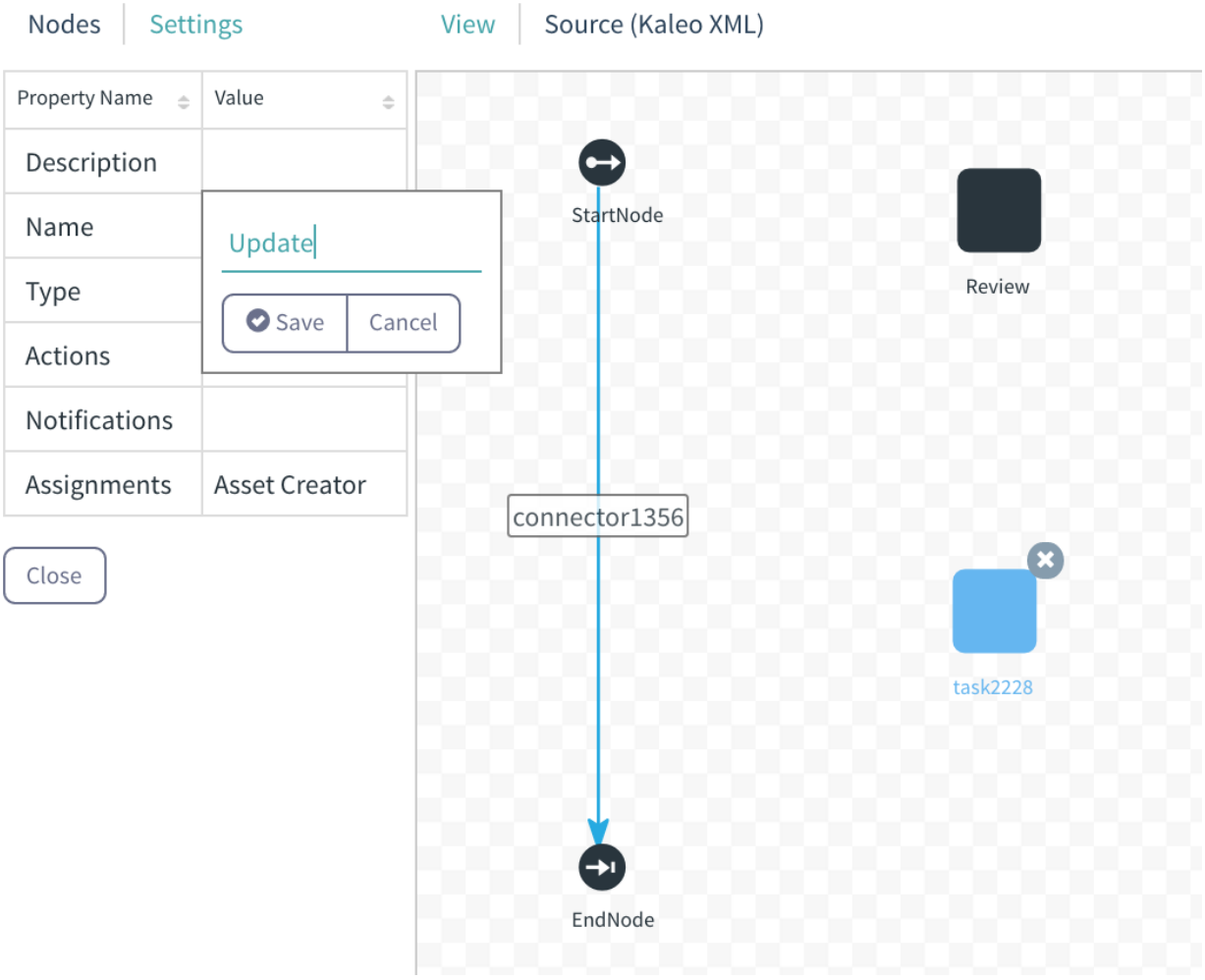


Figure 26.4: You can edit a node's settings.

Fork, Join, and Condition nodes are discussed, along with Transitions, in an article on workflow processing, since they're used for affecting the processing of the workflow.

This article discusses State nodes, Start nodes, and End nodes, as well as Actions and Notifications.

Node Actions and Notifications

Any node can have Actions and Notifications.

Actions Actions are used to do some additional processing before entering the node, after exiting a node, or once a task node is assigned. They're configured by accessing a node's Settings tab, then double clicking *Actions*.

The Single Approver workflow contains an Update task with an action written in Groovy that sets the status of the asset as *denied*, then sets it to *pending*.

```
import com.liferay.portal.kernel.workflow.WorkflowStatusManagerUtil;
import com.liferay.portal.kernel.workflow.WorkflowConstants;
```

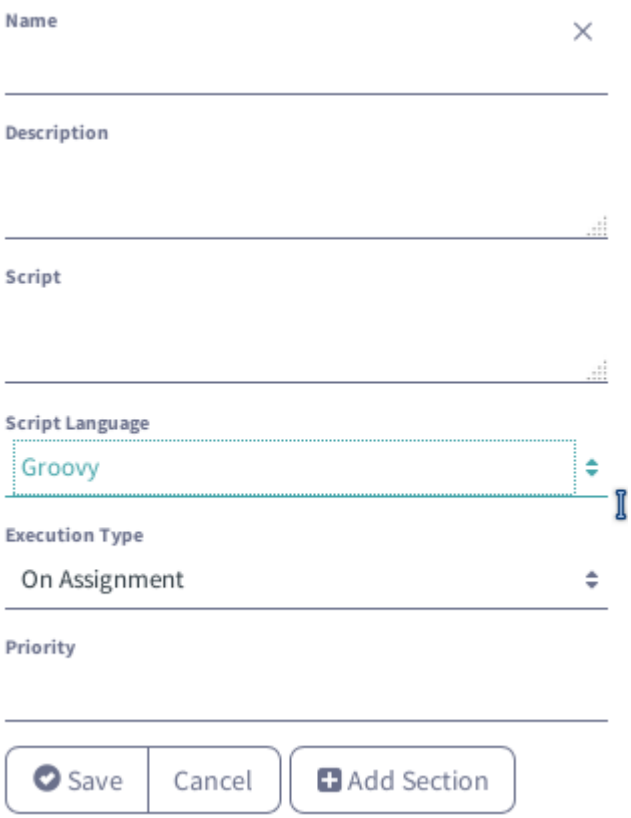


Figure 26.5: You can add an Action to a Task node.

```
WorkflowStatusManagerUtil.updateStatus(WorkflowConstants.getLabelStatus("denied"), workflowContext);
WorkflowStatusManagerUtil.updateStatus(WorkflowConstants.getLabelStatus("pending"), workflowContext);
```

Why would the action script first set the status to one thing and then to another like that? Because for some assets, the *denied* status is used to send the asset creator an email notification that the item has been denied.

The end node in your workflow definition has an action configured on it by default, on entry to the end node:

```
import com.liferay.portal.kernel.workflow.WorkflowStatusManagerUtil;
import com.liferay.portal.kernel.workflow.WorkflowConstants;

WorkflowStatusManagerUtil.updateStatus(WorkflowConstants.getLabelStatus("approved"), workflowContext);
```

This is a Groovy script that just updates the status to *approved*, since that's usually the goal of a workflow process.

You can do something simple like the actions above, or you can be as creative as you'd like.

It's good to assign a task to a user, and it's even more useful if the user can get a notification that something is required in the workflow.

Notifications Notifications are often sent to tell task assignees that something is required of them in the workflow or to update asset creators on the status of the process. They can be sent for tasks or any other type

of node in the workflow. To set up notifications, double click on *Notifications* in a node's Settings tab and create a notification.

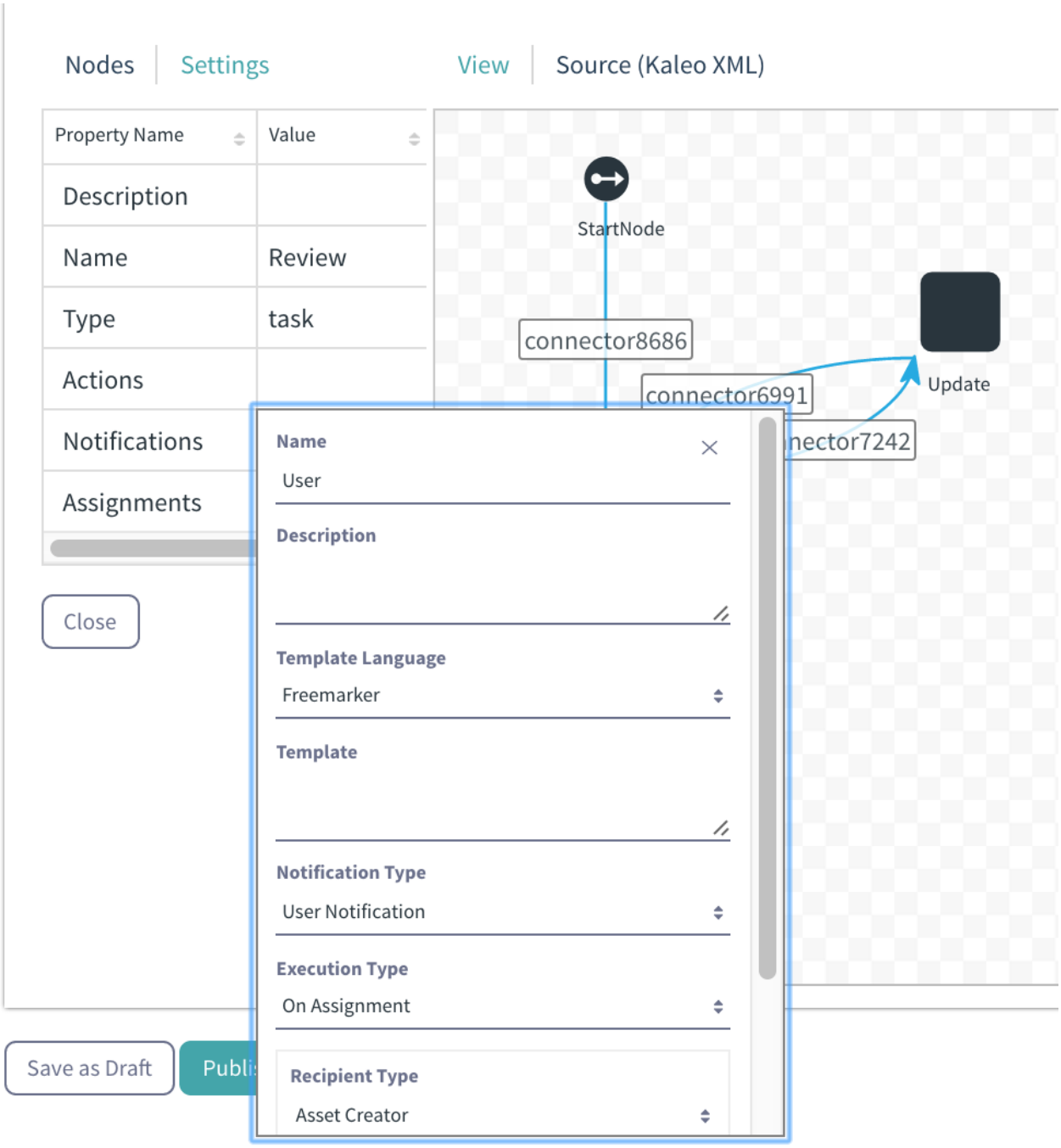


Figure 26.6: You can send notification from a Task node.

You'll need to specify the Notification Type, and you can choose User Notification or Email. You can use Freemarker or Velocity if you need a template, or you can choose to write a plain text message.

Note: Instant Messenger and Private Message also appear as Notification Type options, but these are non-functional and will be removed in a future version.

Here's a basic Freemarker template that reports the name of the asset creator and the type of asset in the notification:

```
${userName} sent you a ${entryType} for review in the workflow.
```

You can also choose to link the sending of the notification to entry into the node (On Entry), when a task is assigned (On Assignment), or when the workflow processing is leaving a node (On Exit). You can configure multiple notifications on a node.

Commonly, the assignment and notification settings are teamed up so a user receives a notification when assigned a task in the workflow. To do this you just choose *Task Assignees* under Recipient Type when configuring the notification.

Start and End Nodes

Start and end nodes kick off the workflow processing and bring the asset to its final, approved state. Often you can use the default start and end nodes without modification. If you want to do some more processing (in the case of a start node), add an action to the node using the Settings tab, as described in the section on Actions above.

End nodes have a default action that sets the workflow status to Approved using the Groovy scripting language:

```
import com.liferay.portal.kernel.workflow.WorkflowStatusManagerUtil;
import com.liferay.portal.kernel.workflow.WorkflowConstants;

WorkflowStatusManagerUtil.updateStatus(WorkflowConstants.getLabelStatus("approved"), workflowContext);
```

Feel free to add more to the action script if you need to do additional processing.

By default, there's a transition connecting the start node and end node, but you'll probably want to delete it, since most workflows won't proceed straight from the initial state to approved.

State Nodes

State nodes can have Notifications and Actions. The default end node added by Kaleo Designer is a pre-configured state node that sets the workflow status to Approved. Perhaps you want to create a node that sets the status to *Expired*. You could create a state node for it by dragging one onto your Kaleo Designer canvas, then configuring an action in it that sets the status to Expired. Here's what it would look like in Groovy:

```
import com.liferay.portal.kernel.workflow.WorkflowStatusManagerUtil;
import com.liferay.portal.kernel.workflow.WorkflowConstants;

WorkflowStatusManagerUtil.updateStatus(WorkflowConstants.getLabelStatus("expired"), workflowContext);
```

Next, you'll learn to do parallel processing using fork and join nodes.

Related Topics

Workflow Forms

Using Workflow

Liferay's Workflow Framework

Creating Simple Applications

26.3 Affecting the Processing of Workflow Definitions

Workflow definitions all contain nodes: a Start Node, and End node, and at least one Task node. You might know that for the workflow to progress from one node to the other, you need Transitions. In this article you'll learn about using transitions to move the asset through the workflow from node to node, but you'll also learn about some other features that affect the processing of the workflow.

- Transitions
- Forks
- Joins
- Conditions

Start by learning about the ever important Transition.

Transitions

What are transitions? Workflow transitions connect one node to another. On exiting the first node, processing continues to the node pointed to by the transition. Every time you create an arrow from one node to another, Kaleo Designer creates a transition.

Each node you add has a pop-up menu letting you delete the node. As you hover your mouse over the edges of a node, notice your mouse pointer changes to a cross. The cross indicates you can connect the current node to another node. Hold down your mouse button and drag the mouse to start drawing your transition to another node. If you stop before reaching the edge of the next node, a pop-up displays node types you can create and connect to on-the-fly. To connect with an existing node, continue dragging the connector to that node.

When developing workflows in Kaleo Designer, make sure you go through all the transitions and name them appropriately. By default, these transitions get system generated names, so rename them all to something more human-readable, as they're displayed to workflow users as links that will send the workflow item to the next step in the workflow.

To rename transitions, click on the arrow representing the transition and use the Setting tab to set the name just like you do for a node.

Forks and Joins

Sometimes you don't need to wait for one task to be completed before moving on to another one. Instead, you want to do two or more things at the same time. To do this, transition to a fork node, make two transitions from the fork to your parallel tasks, and then come back together using a join node.

With a regular Join node, for the workflow to proceed beyond the join, the transition from both parallel executions must be invoked. However, if you use a Join XOR node instead, the workflow proceeds as long as the transition from one of the parallel executions is invoked.

Keep in mind that you must balance your fork and join nodes. In other words, for every fork, there must be a join that brings the parallel workflow threads back together.

Conditions

Sometimes you need to inspect an asset or its execution context, and depending on the result, send it to the appropriate transition. You need a node for a script that concludes by setting a value to one of your transitions.

From the *Category Specific Approval* (`category-specific-definition.xml`), this is the script in the condition node that starts the workflow (coming directly from the start node):

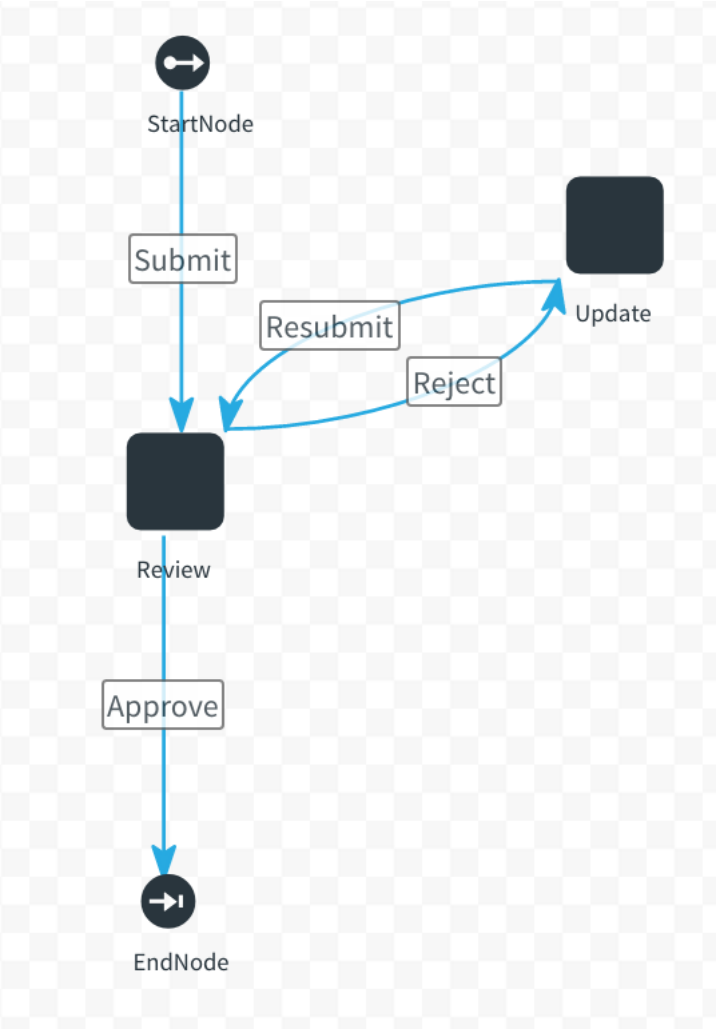


Figure 26.7: You connect nodes and direct workflow processing with transitions. The Single Approver workflow has transitions named Submit, Resubmit, Reject, and Approve.

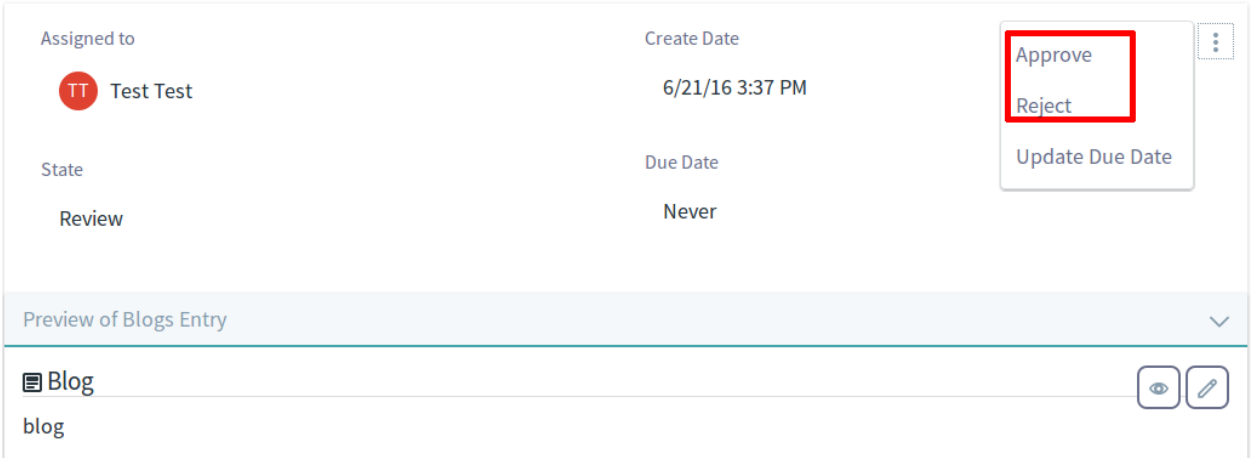


Figure 26.8: In the Single Approver workflow, a user in the Review task can choose to Approve or Reject the asset, which sends the asset either to the EndNode or to the Update task.

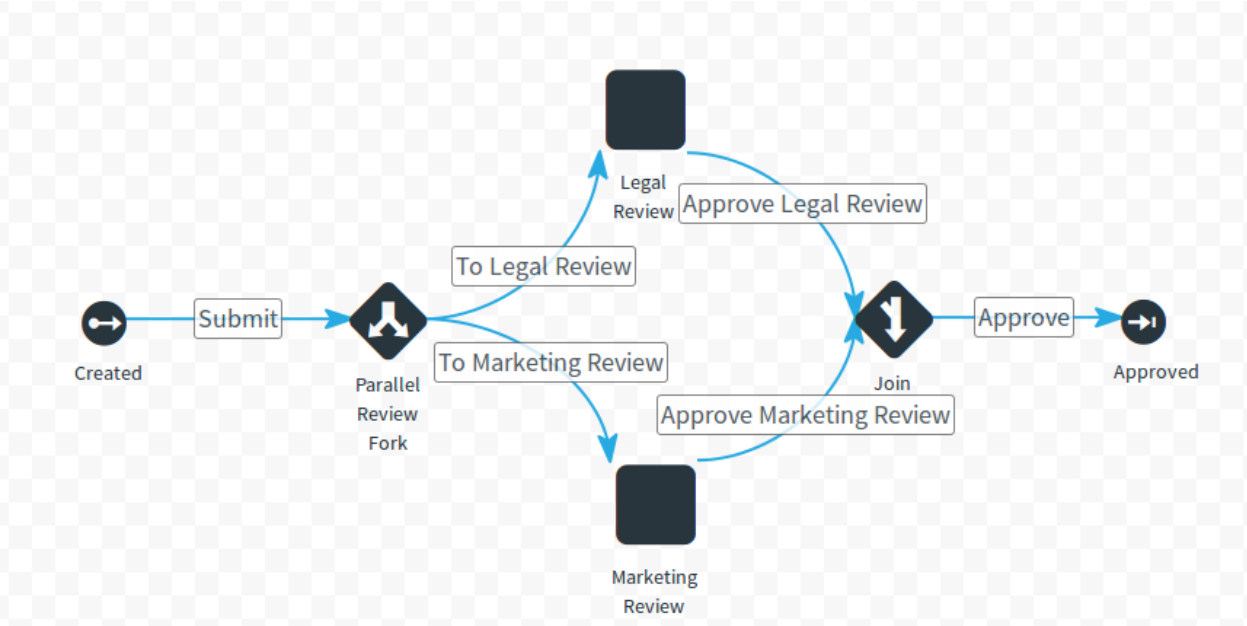


Figure 26.9: Forks and Joins are used to enable parallel processing in the workflow.

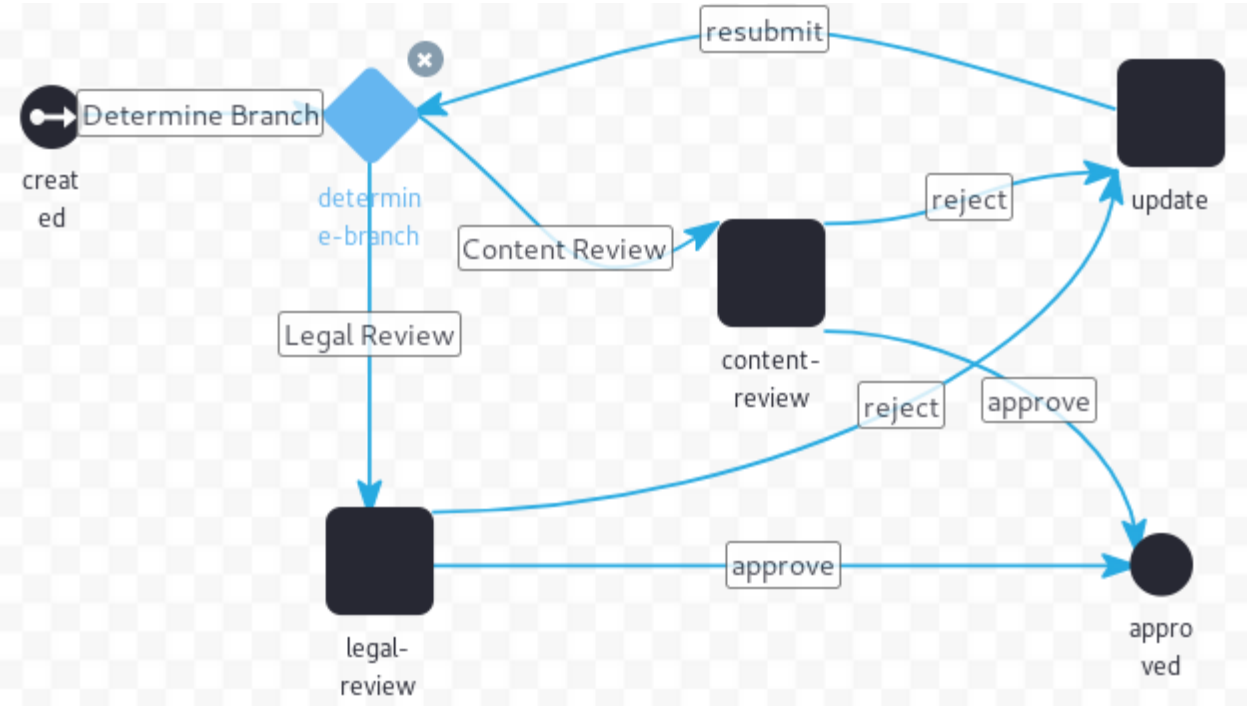


Figure 26.10: The Category Specific Approval definition starts with a Condition node.

```
import com.liferay.asset.kernel.model.AssetCategory;
import com.liferay.asset.kernel.model.AssetEntry;
import com.liferay.asset.kernel.model.AssetRenderer;
import com.liferay.asset.kernel.model.AssetRendererFactory;
import com.liferay.asset.kernel.service.AssetEntryLocalServiceUtil;
import com.liferay.portal.kernel.util.GetterUtil;
import com.liferay.portal.kernel.workflow.WorkflowConstants;
import com.liferay.portal.kernel.workflow.WorkflowHandler;
import com.liferay.portal.kernel.workflow.WorkflowHandlerRegistryUtil;

import java.util.List;

String className = (String)workflowContext.get(WorkflowConstants.CONTEXT_ENTRY_CLASS_NAME);

WorkflowHandler workflowHandler = WorkflowHandlerRegistryUtil.getWorkflowHandler(className);

AssetRendererFactory assetRendererFactory = workflowHandler.getAssetRendererFactory();

long classPK = GetterUtil.getLong((String)workflowContext.get(WorkflowConstants.CONTEXT_ENTRY_CLASS_PK));

AssetRenderer assetRenderer = workflowHandler.getAssetRenderer(classPK);

AssetEntry assetEntry = assetRendererFactory.getAssetEntry(assetRendererFactory.getClassName(), assetRenderer.getClassPK());

List<AssetCategory> assetCategories = assetEntry.getCategories();

returnValue = "Content Review";

for (AssetCategory assetCategory : assetCategories) {
    String categoryName = assetCategory.getName();

    if (categoryName.equals("legal")) {
        returnValue = "Legal Review";

        return;
    }
}
```

This example checks the asset category to choose the processing path, whether to transition to the *Legal Review* task or the *Content Review* task.

You may be wondering what that `returnValue` variable is. It's the variable that points from the condition to a transition, and its value must match a valid transition in the workflow definition. This script looks up the asset in question, retrieves its asset category, and sets an initial `returnValue`. Then it checks to see if the asset has been marked with the *legal* category. If not it goes through *Content Review* (the content-review task in the workflow), and if it does it goes through *Legal Review* (the legal-review task in the workflow).

Now you're equipped with the basic knowledge to design beautiful, effective workflows so that your assets can be properly reviewed before they're published in your sites.

Related Topics

Workflow Forms
- Using Workflow
- Liferay's Workflow Framework
- Creating Simple Applications

26.4 Creating Tasks in Kaleo Designer

Task nodes have several parts and are the most complex parts of a workflow definition. Unlike other workflow nodes, task nodes have Assignments, because a user is expected to *do something* (often approve or reject the

submitted asset) when a workflow process enters the task node: the assignment specifies who that user is.

Commonly, task nodes contain Notifications, Assignments, and Actions (defined in scripts). See more about Notifications and Actions in the article on workflow nodes. Task nodes and their assignments are more complex and deserve their own article (this one).

To get started, drag and drop a task node on your workflow canvas if you haven't already. Open its settings and give it a name as described above. Then double click *Actions* in the task's Settings pane.

You can define a notification (often Task Assignee is appropriate), or write a Groovy script defining an action that's triggered for your task.

Next learn about creating Assignments for your task nodes.

Assignments

Workflow tasks need to be completed by a Liferay DXP user. You can choose how you want to configure your assignments.

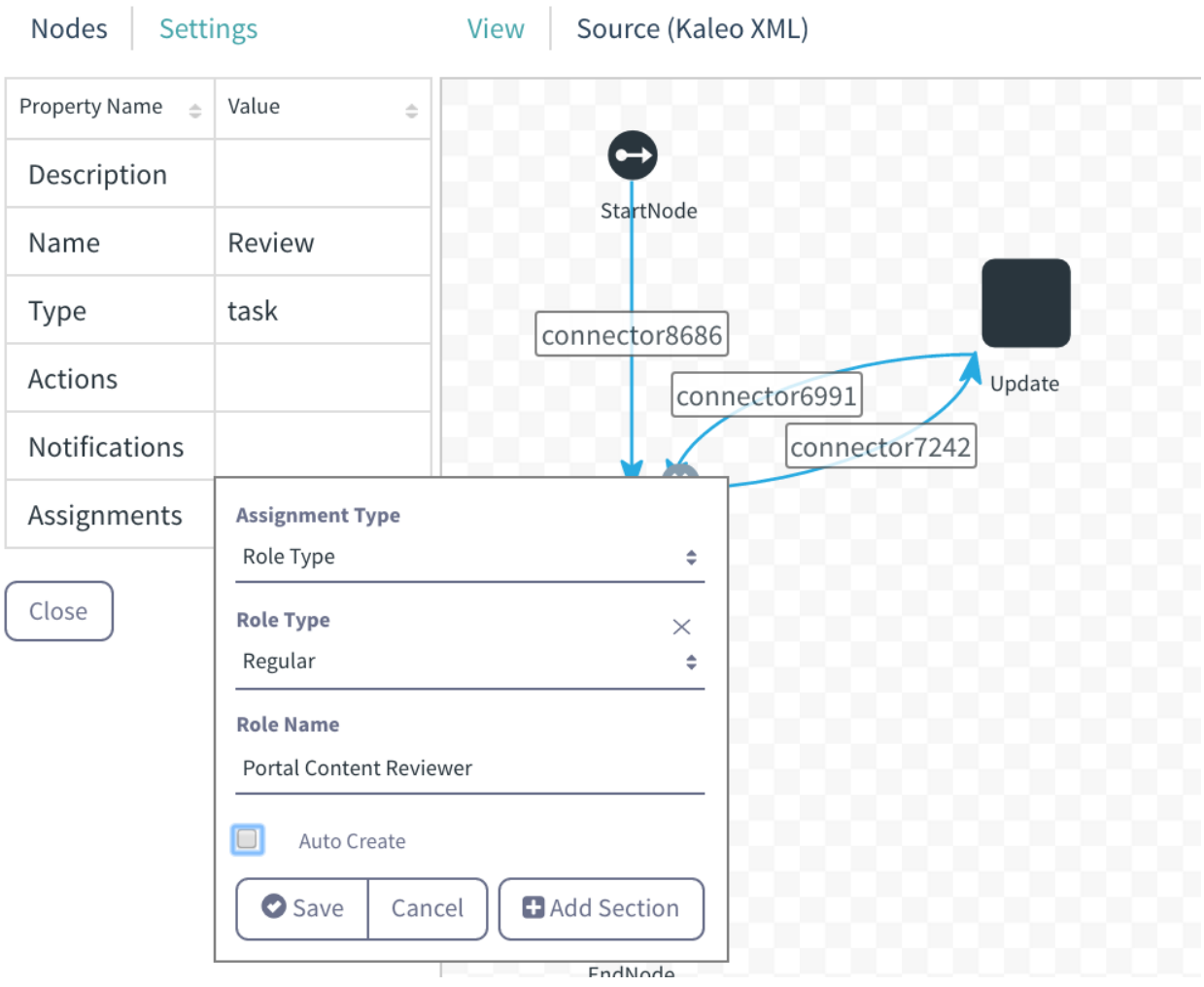


Figure 26.11: You can add an Assignment to a Task node.

You can choose to add assignments to specific Roles, multiple roles of a Role Type (organization, site, or regular role types), to the Asset Creator, to Resource Actions, or to specific Users. Additionally, you can write a script to define the assignment.

Assigning tasks to Roles, Organizations, or Asset Creators is a straightforward concept, but what does it mean to assign a workflow task to a Resource Action? Imagine an *UPDATE* resource action. If your workflow definition specifies the UPDATE action in an assignment, then anyone who has permission to update the type of asset being processed in the workflow is assigned to the task. You can configure multiple assignments for a task.

Resource Action Assignments *Resource actions* are operations performed by users on an application or entity in Liferay DXP. For example, a user might have permission to update Message Boards Messages. This is called an UPDATE resource action, because the user can update the resource. If you're still uncertain about what resource actions are, refer to the developer tutorial on Liferay DXP's permission system for a more detailed explanation.

To find all the resource actions that have been configured in Liferay DXP, you need access to the Roles Admin application in the Control Panel (in other words, you need permission for the VIEW action on the roles resource).

- Navigate to Control Panel → Users → Roles.
- Add a new Regular Role. See the article on managing roles for more information.
- Once the role is added, navigate to the Define Permissions interface for the role.
- Find the resource whose action you want to use for defining your workflow assignment.

How do you go from finding the resource action to using it in the workflow? If you're running Liferay DXP Fix Pack 13 or greater, use Kaleo Designer's interface for setting up a resource action assignment.

When configuring your task node's Assignment, select Resource Actions as the Assignment Type, then specify the Resource Actions to use for the assignment (for example, UPDATE).

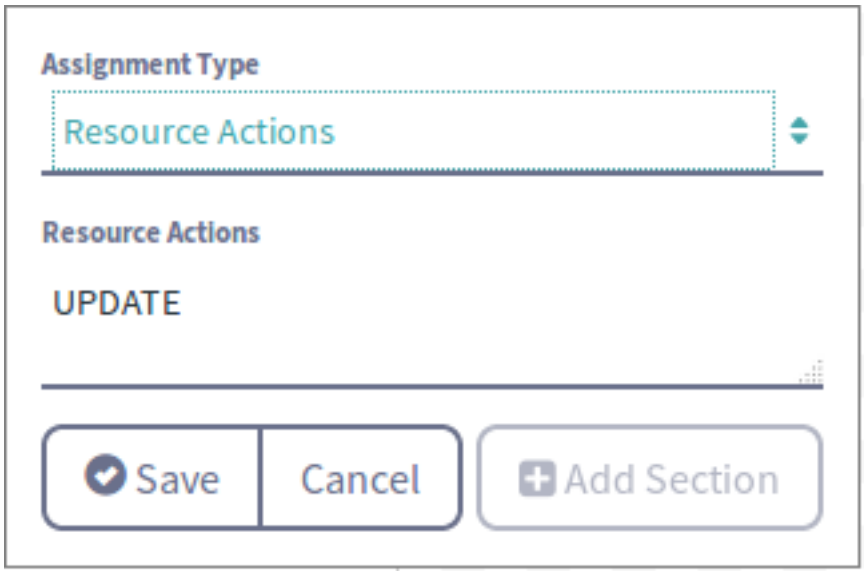


Figure 26.12: Configure resource action assignments in Kaleo Designer.

Here's what the assignment looks like in the Source (Kaleo XML) tab:

```
<assignments>
    <resource-actions>
        <resource-action>UPDATE</resource-action>
    </resource-actions>
</assignments>
```

As usual, assign the workflow to the appropriate workflow enabled asset.

Now when the workflow proceeds to the task with the resource action assignment, users with `UPDATE` permission on the resource (for example, Message Boards Messages) is notified of the task and can assign it to themselves (if the notification is set to Task Assignees). Specifically, users see the tasks in their *My Workflow Tasks* application under the tab *Assigned to My Roles*.

Use all upper case letters for resource action names. Here are some common resource actions:

```
UPDATE
ADD
DELETE
VIEW
PERMISSIONS
SUBSCRIBE
ADD_DISCUSSION
```

You can determine the probable resource action name from the permissions screen for that resource. For example, in Message Boards, one of the permissions displayed on that screen is *Add Discussion*. Convert that to all uppercase and replace the space with an underscore, and you have the action name.

Scripted Assignments You can also use a script to manage the assignment. Here's the script for the Review task assignment in the Scripted Single Approver workflow definition:

```
import com.liferay.portal.kernel.model.Group;
import com.liferay.portal.kernel.model.Role;
import com.liferay.portal.kernel.service.GroupLocalServiceUtil;
import com.liferay.portal.kernel.service.RoleLocalServiceUtil;
import com.liferay.portal.kernel.util.GetterUtil;
import com.liferay.portal.kernel.workflow.WorkflowConstants;

long companyId = GetterUtil.getLong((String)workflowContext.get(WorkflowConstants.CONTEXT_COMPANY_ID));

long groupId = GetterUtil.getLong((String)workflowContext.get(WorkflowConstants.CONTEXT_GROUP_ID));

Group group = GroupLocalServiceUtil.getGroup(groupId);

roles = new ArrayList<Role>();

Role adminRole = RoleLocalServiceUtil.getRole(companyId, "Administrator");

roles.add(adminRole);

if (group.isOrganization()) {
    Role role = RoleLocalServiceUtil.getRole(companyId, "Organization Content Reviewer");

    roles.add(role);
}
else {
    Role role = RoleLocalServiceUtil.getRole(companyId, "Site Content Reviewer");

    roles.add(role);
}

user = null;
```

Don't let all that code intimidate you. It's just assigning the task to the *Administrator* role, then checking whether the *group* of the asset is an Organization and assigning it to the *Organization Content Reviewer* role if it is. If it's not, it's assigning the task to the *Site Content Reviewer* role.

Note the `roles = new ArrayList<Role>();` line above. In a scripted assignment, the `roles` variable is where you specify any roles the task is assigned to. For example, when `roles.add(adminRole);` is called, the Administrator role is added to the assignment.

Related Topics

Workflow Forms
- Using Workflow
- Liferay's Workflow Framework
- Creating Simple Applications

26.5 Workflow Forms

Business processes are often form-based and workflow-driven. They start with some data that needs to be entered and progress by sending the data to other people or groups. The initial data is processed in some way (for example, further data is entered or approval is granted), and the process moves on until completion, when each interested party has seen and manipulated the data. To write an app for each of these fluid processes is laborious and not cost effective. Instead, a tool for quickly defining a process to suit each use case is needed. The process architect must define the data that gets collected and define the process the data must move through to reach its final state. To accomplish this, Liferay DXP already includes the Dynamic Data Lists application for defining forms, and the Kaleo Designer application for designing workflows. The Kaleo Forms solution combines the features of these applications, letting you design an integrated process for sending forms through a workflow in one intuitive UI.

The entire process includes:

1. Creating a Kaleo Process
2. Creating the process's field set (or selecting an existing Data Definition)
3. Creating the process's workflow definition (or selecting one that's already created)
4. Creating and assigning forms for each workflow task

Example Use Case:

The Lunar Resort spa is a popular place. With all the adventure-driven activities (lunar hiking, rover races, etc.) available to guests, it's important to offer some relaxing and rejuvenating experiences. As the spa expands and new Spa Technicians are hired, the Spa Manager decides to implement a new process to handle orders. It involves several steps and several people:

- The customer service representative for the spa interacts with a customer and fills out an order form.
- The customer service representative gives the order details to the spa technicians so they can divide up the requests, with a single technician assigned to each order.
- The spa technician sends the order to the spa manager for final approval and comment.

- The spa manager sends the final order back to the customer service representative, who contacts the customer with the final details of his spa service order.

A process like this is a nightmare to manage manually, but perfectly suited for Kaleo Forms.

Since Kaleo Forms is accessed through a site's administration menu (under Content → Kaleo Forms Admin), make sure the intended users of your Kaleo Processes have access to the application. See the article on roles and permissions if you're unsure how that works.

Kaleo Forms Display was removed from 7.0: In prior versions of Liferay Portal EE (for example, see here), A Kaleo Forms Display portlet could be added to a site page, so users with access to the page and the proper permissions could add, delete, and submit Kaleo Processes. No capabilities have been removed from the Kaleo stack in 7.0, but the Kaleo Forms Display portlet was removed. This means that administrators must grant permission for users to access Kaleo Forms in Site Administration. Users needing only to manage workflow tasks assigned to them or their roles can use the *My Workflow Tasks* section of their personal sites (in the User Menu, go to *My Account* → *My Workflow Tasks*).

Using Kaleo Forms

Access Kaleo Forms from the *Content* section of your site's administration menu (click *[Site Name]* → *Content* → *Kaleo Forms Admin*). Kaleo Forms appears with a listing of the defined processes. If you're coming here for the first time, there won't be any, so create one.

Creating a Kaleo Process Click the *Add* ( ) button to open the New Process Wizard and begin the first step in creating a Kaleo Process. Name the process *Spa Order Process*, add a description, and then click *Next*.

Creating the Form In the second step of the New Process Wizard, define the fields that can appear in your process's forms. There are two methods:

1. Choose a predefined field set. To use a field set that already exists, pick one from the list of field sets by clicking the Kebab menu ( ) next to its entry and selecting *Choose*.
2. Create a new field set. Use the *Add Field Set* button to add a new field set. If you're not sure about creating a field set, see the documentation on creating data definitions first.
    - For this example, Add a new field set:
        - A Text field called *Customer Name*
        - A Text Box called *Requested Spa Technician*
        - A Select field called *Services Requested* with these options:
            * Massage
            * Fish Pedicure
            * Sauna
            * Acupuncture
            * Moon Mud Facial Mask
        - A Boolean field called *Returning Customer?*
        - A Date field called *Preferred Date*
        - A Text field called *Preferred Time*
        - A Date field called *Available Date*

Figure 26.13: Add a Kaleo Forms Process to link a form with a workflow definition.

- A Text field called *Available Time*
- A Boolean field called *Approved*
- A Text Box field called *Managerial Comments*

Click *Next* to move to the next step in the wizard: adding a workflow for the process.

Choosing a Workflow In the third step of the New Process Wizard you select a workflow to use for your forms. The Single Approver workflow is included by default. See how it's defined by opening it in the Kaleo Designer. Click the kebab menu (⋮) and then *Edit*.

The graphical interface for editing or defining a workflow appears below the Single Approver edit screen's Details section.

If you don't want to base your workflow on the Single Approver definition (or any definition already available in your system), add a new workflow from scratch using the Kaleo Designer. Click the *Add Workflow* button to get started.

- For the Spa Order Process, make a new workflow definition called *Spa Order Workflow*:

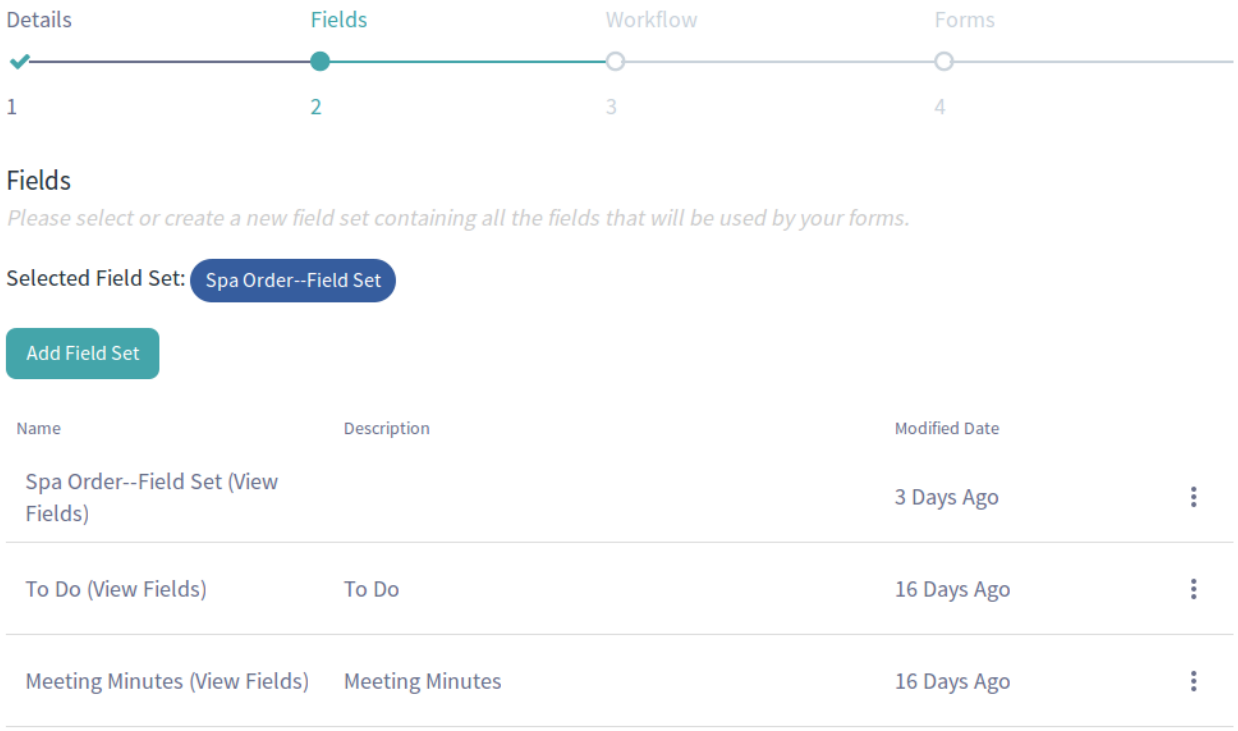
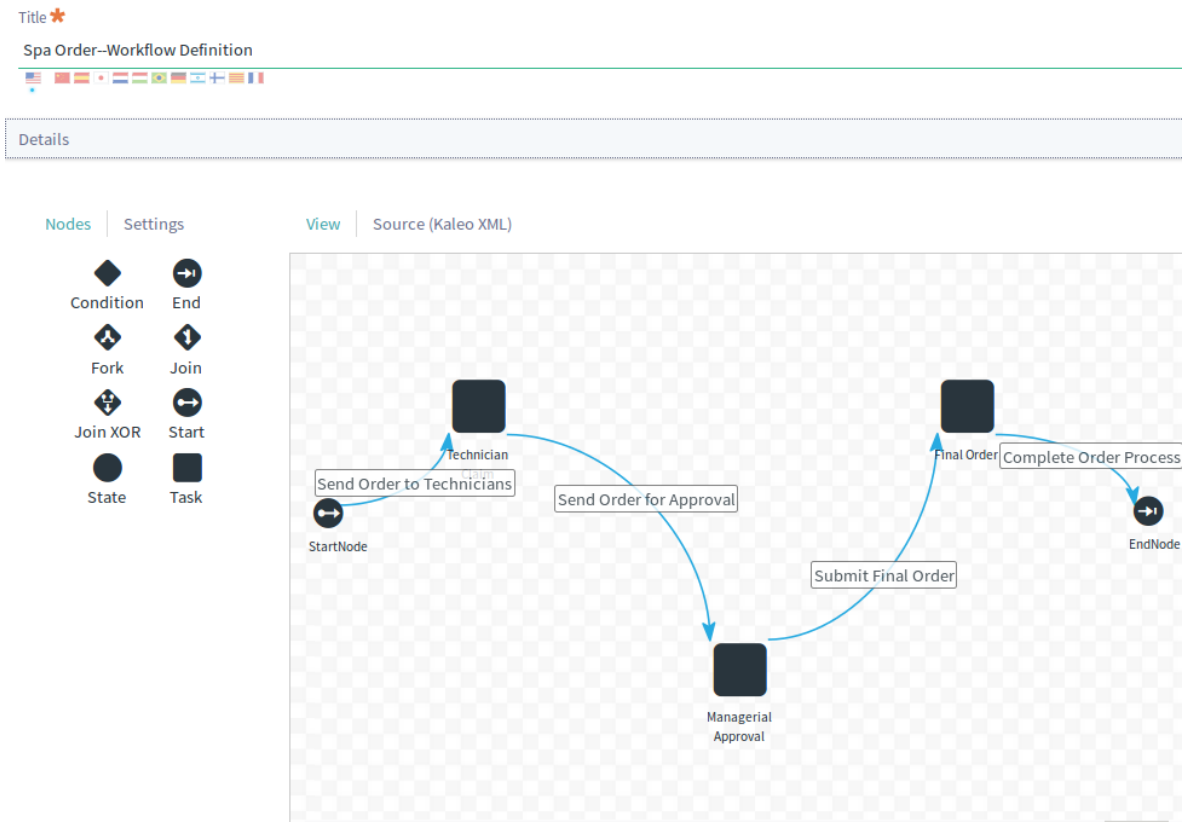


Figure 26.14: In the second step of the New Process Wizard, define and choose the fields for your form.



The definition has a StartNode and three task nodes. Develop it in the graphical designer built into Kaleo Forms. There's a more detailed write-up of the Kaleo Designer here if you're not already familiar with it.

This workflow exhibits simple linear processing, so that the initial form is filled out (in the StartNode) and then the workflow moves to the first task (Technician Claim), then to the second task (Managerial Approval), and finally, to the final order task.

The task assignments of this workflow are as follows:

- Technician Claim: Assigned to the Spa Technician Role.
- Managerial Approval: Assigned to the Spa Manager Role.
- Final Order: Assigned to the Asset Creator.

Note: Create the Spa Technician Role and the Spa Manager Role as Site Roles, and give them permission to access the Kaleo Forms application. In the role's page for defining permissions, they're under Site Administration → Content → Kaleo Forms Admin. Give both roles all of the permissions available for Kaleo Forms Admin.

Once the workflow is done, select it: back on the third step of the New Process wizard, click *Actions* → *Choose* on the workflow you just created. Then click *Next*.

Adding Task Forms In the fourth (and final) step of the New Process wizard, assign a form to each task in the workflow definition (and to the StartNode to get the process kicked off). All forms you work with here will be either the entire field set you defined in step two, or a subset of it.

Note: In this step, you can't add any proper form fields (fields that accept user input). If you forgot to add a Select field to the field set in step two, for example, you cannot add one here. Instead go back and add it to the field set.

While you can't add form fields, you can configure the existing fields differently for each form, and you can add these elements to the form that enhance its usability:

1. Add a *Fieldset* to provide headings in a form. For example, name it after the workflow task to provide clarity to the task assignee.
2. Use a *Paragraph* field to add a block of text to the form. For example, provide instructions to assignees about what's expected of them to complete the task.
3. Use the *Separator* to separate sections of the form visually. For example, combine a Paragraph and a Separator to instruct the assignee to focus on a certain section of the form.

For review, the Spa Order Process has these fields:

- Customer Name
- Requested Spa Technician
- Services Requested
- Returning Customer?
- Preferred Date
- Preferred Time
- Available Date
- Available Time

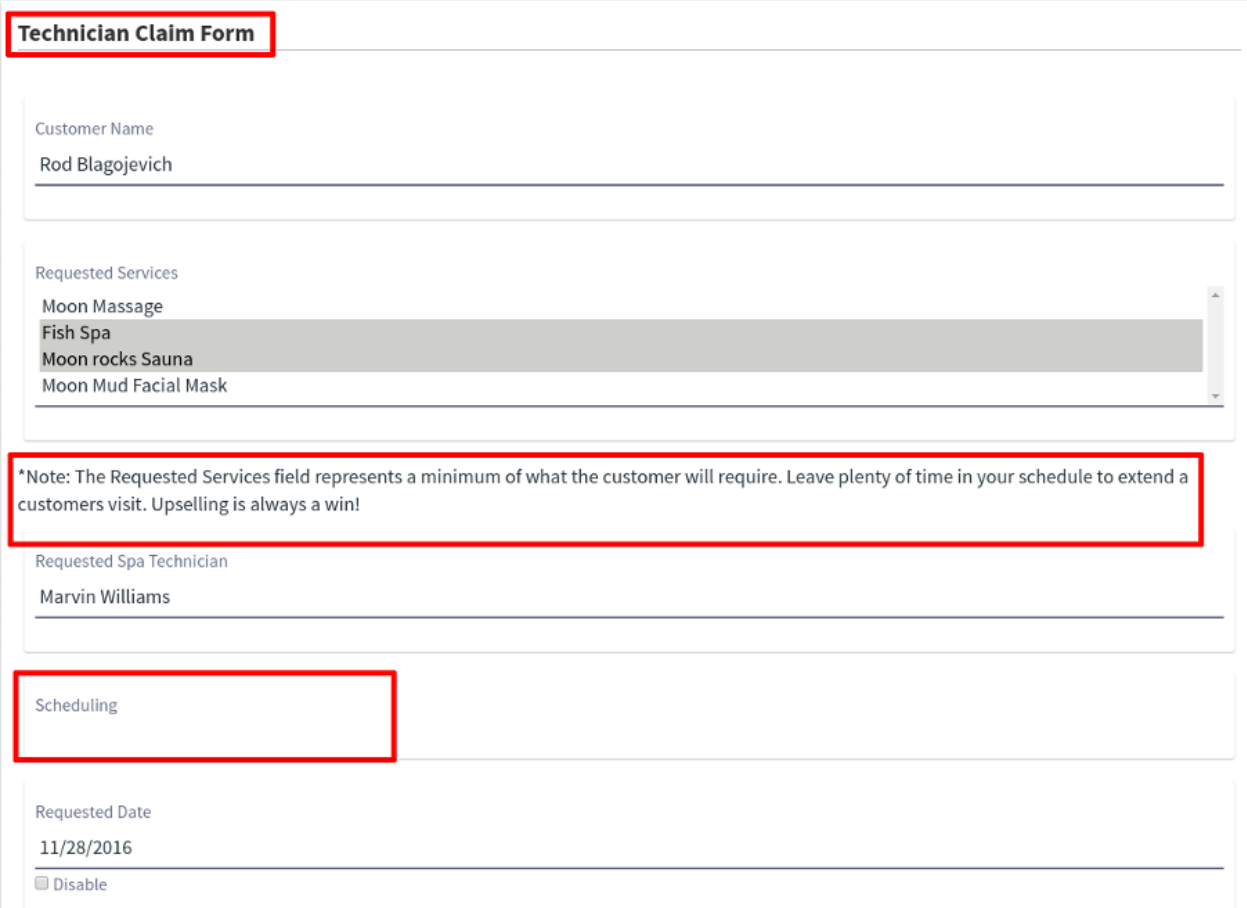


Figure 26.15: Use Fieldsets, Paragraphs, and Separators to enhance the appearance of your workflow task forms.

- Approved
- Managerial Comments

Not all the available fields should appear on each form. For example, the Approved field should only appear on the form for the managerial review task.

Each applicable workflow node (the initial state and each task) appears in a row with its associated form and a button that lets you assign a form. In the Spa Order Workflow definition, there are four nodes that need forms: StartNode, Technician Claim, Managerial Approval, and Final Order.

Click on the kebab menu (⋮) next to the Request Entered task and then click *Assign Form*. The screen that appears lets you create a new form if you click the *Add* button ( ).

- First, name the initial form (call it *Spa Order Process–Order Form* if you like).
- Next look at the available fields. In this stage of the workflow, you don't need all the fields. Delete these fields from the form (mouse over the field and click the delete icon–):
  - Available Date
  - Available Time
  - Approved
  - Managerial Comments

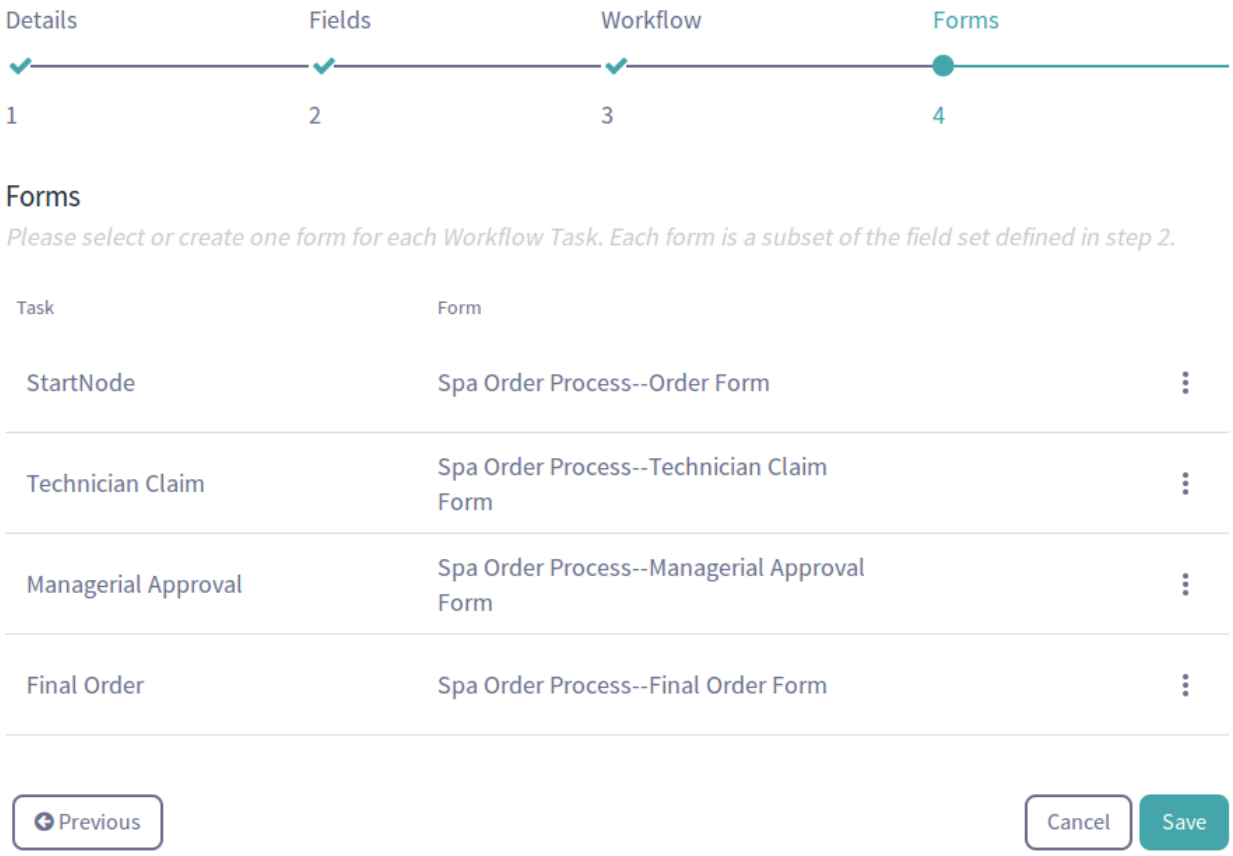


Figure 26.16: You can assign a form to each task in the workflow, and for the initial state (*created* in this case.)

Note: If you delete a field by accident or want to add another field to your form, drag and drop its icon from the *Fields* tab on the left to the form canvas on the right. Fields already on the form that can appear on it only once are grayed out. You can also edit the settings of a field in the form by mousing over it and clicking the wrench icon.

If you don't need to change any field settings in the form, click *Save*. On the next screen choose your new form by clicking on it. Now the workflow definition's StartNode has a form associated with it.

Next create a form named *Spa Order Process–Technician Claim Form* for the Technician Claim workflow task. Follow the same procedure used above, this time deleting these fields from the full field set:

- Approved
- Managerial Comments

Example Use Case Note: In the Spa Order Process, the Managerial Approval workflow task and its associated form are filled out by the spa manager. The manager can oversee the order, making sure that customers requests are being met and that all technicians are fulfilling their share of the spa's duties.

Create a form to be used in the Managerial Approval task, called *Spa Order Process–Managerial Approval Form*. The manager can change anything she wants about the order, so include the entire field set in the form.

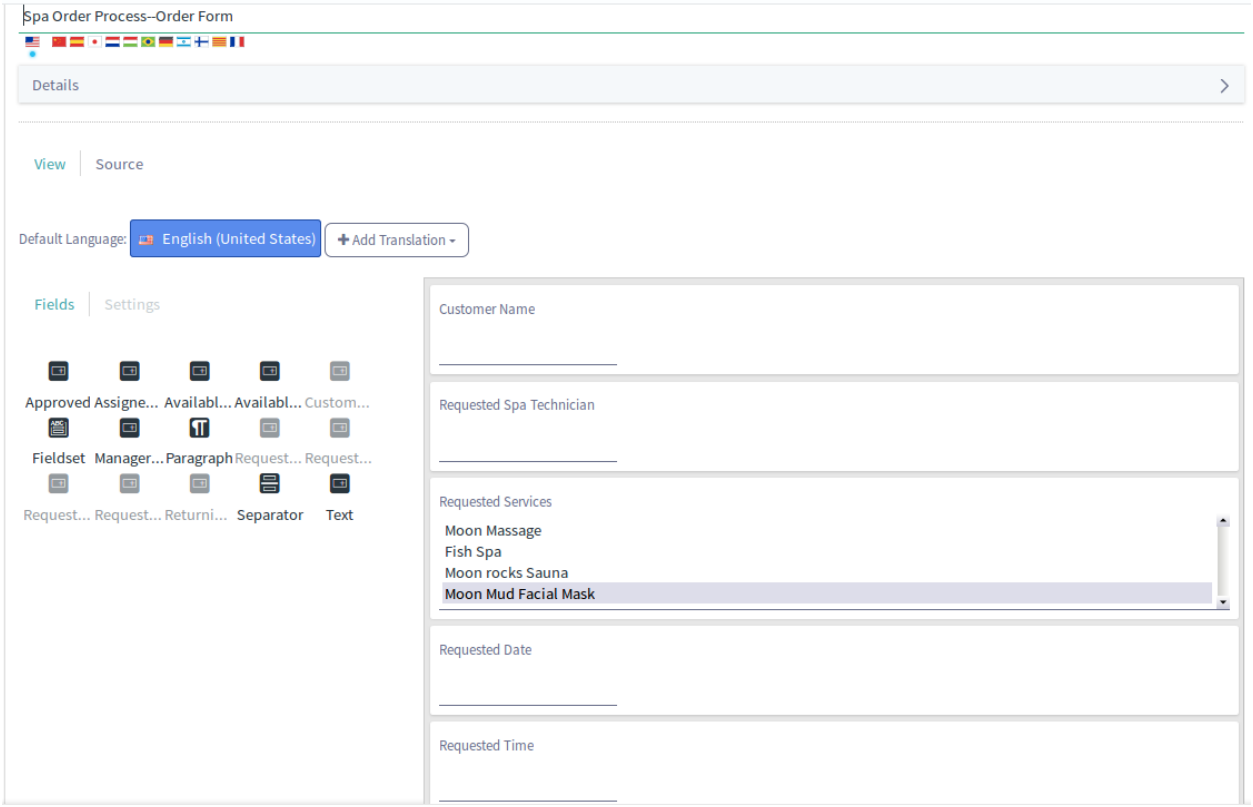


Figure 26.17: Workflow task forms are a subset of the original form.

Example Use Case note: The customer service representative finalizes the order and gets back to the customer on the final details (available time, name of technician, and confirms the services they want).

There's one more form to create: the *Spa Order Process–Final Order Form*. Like the last form, it should have all the form's fields so that the customer service representative can review the order with the customer.

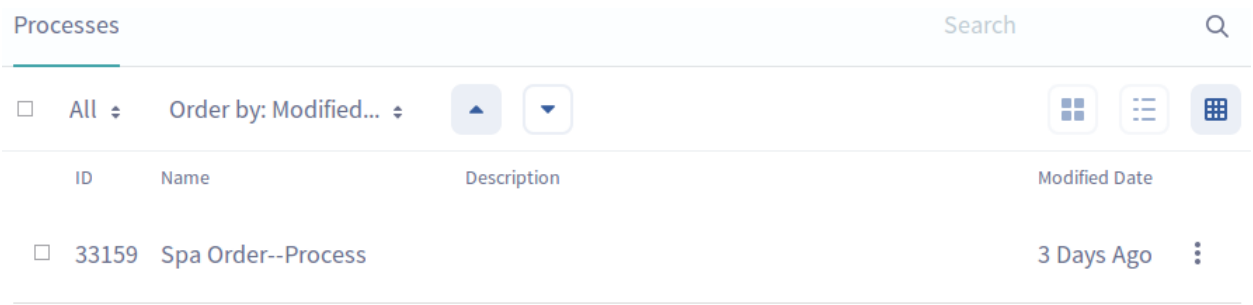


Figure 26.18: Once created, your process is listed in Kaleo Forms.

Click *Save* when finished. The process is ready to test.

Test the Process Once your new process is developed, add records to it by clicking on the process name and then the Plus button ( ).

The application displays the form you assigned to the workflow's initial state. Fill out the form and click Save.

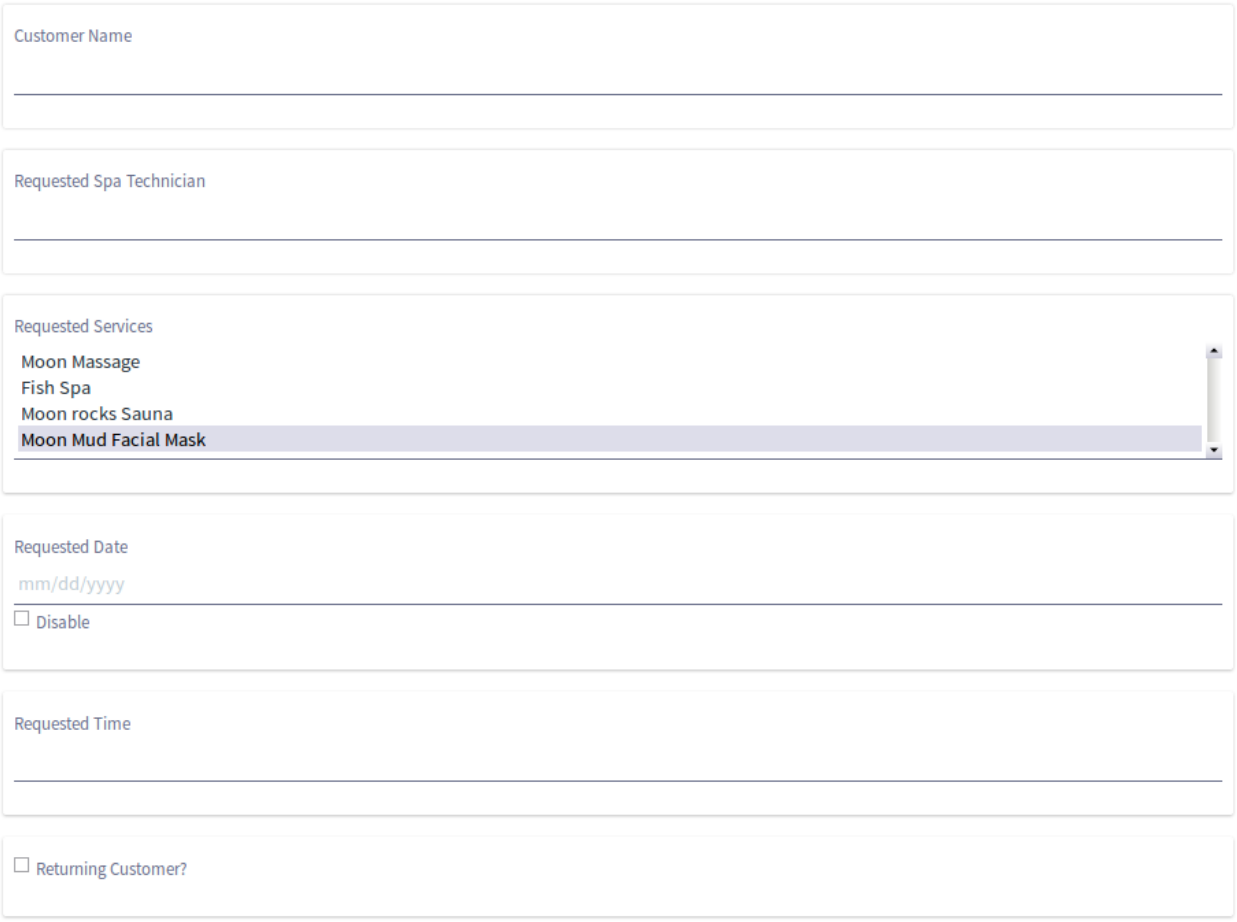


Figure 26.19: Fill out the form and send it through the process.

Once the initial form is submitted in the Kaleo Forms application, the process is managed by the workflow engine. In a process backed by the Spa Order Workflow definition, the workflow moves first to the Technician Claim task, then to the Managerial Approval task, and last, to the Final Order task. Whatever users or roles you assigned to the tasks receive notifications, and the task appears in the Assigned to My Roles section of the My Workflow Tasks portlet. A notification is also triggered in the Notifications application. Once in the task, the user views and approve the form or clicks the edit ( ) button. At this point the workflow task forms you created come into play. Each assigned user fills out the form, saves it, and sends it along in the workflow.

The Kaleo Forms application unites two powerful features of Liferay by integrating a workflow and a form into a new entity, the Kaleo Process.

Related Topics

Kaleo Designer
- Using Workflow
- Liferay's Workflow Framework
- Creating Simple Applications

Chapter 27

User Management

You know how all these retailers advertise themselves as a "one stop shop" for anything you want? The idea is they have so much stuff that whatever you're looking for is probably there. Liferay's Control Panel is something like this. If you want to create users, organizations, sites, configure permissions and plugins and pretty much anything else, you can do it from the Control Panel. When signed in to Liferay as an administrator, you can access the Control Panel from the Product Menu by clicking *Control Panel*.

The Control Panel is divided into five main areas: Sites, Users, Apps, Configuration, and System. The Users section lets you create and manage users, organizations, user groups, roles, and password policies. If monitoring has been enabled for your portal, you can also view all of the live portal sessions of your users from this area of the Control Panel. If the Audit plugins have been installed from Liferay Marketplace, the Audit section also appears.

The Users section of the Control Panel is used for most administrative tasks involving user management. There, you'll find an interface for the creation and maintenance of the following portal entities: users, organizations, user groups, and roles.

Managing sites, teams, site templates, and page templates is covered in detail elsewhere. Remember that it's possible and sometimes simpler to use sites, site memberships, and teams to organize users and manage permissions than it is to use organizations, user groups, and custom roles.

As a portal administrator, you'll use the Users section of the Control Panel to create users, organizations, and user groups, implement security via roles and permissions, and administer your users. Only users with the administrator role, which is a portal scoped role, have permission to view the Control Panel. You can, however, grant permissions to view the Control Panel to existing or custom roles.

In this set of articles you'll learn about managing users. While you could probably get started learning about managing users in Liferay just by perusing the Control Panel, there's so much User Management functionality in Liferay that it's tough to wrap your head around it all. Maybe a limerick can help:

*There's a company in Spain that makes Hula Hoops They placed **Users** in **Organizations** and **User Groups** They assigned them their **Roles** And set them some goals Made **Password Policies** that can't be duped.*

Begin exploring Liferay's User Management functionality by reading about adding and editing users.

27.1 Adding and Managing Users

Users are a fundamental entity in Liferay. If your portal requires people (even just a set of site administrators) to have an account that they sign into to do anything in the portal, you need to know about users. If your

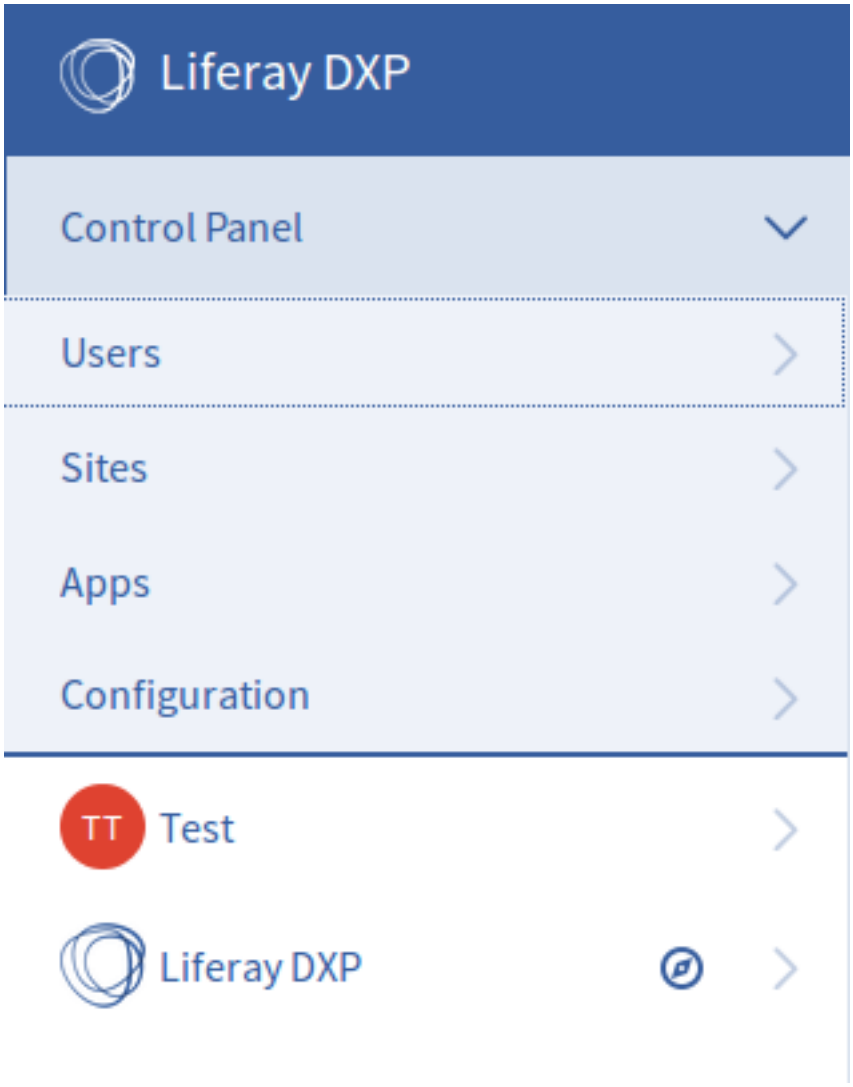


Figure 27.1: Administrators can access the Control Panel from the Product Menu.

users are at all divided hierarchically, like into departments, you might find Organizations helpful. See the article on organizations for more information.

Whether or not you've ever used Liferay before, you're probably not surprised to hear that Users are managed in the Control Panel's *User* section. If it were any different, it'd be weird.

For the Lunar Resort example portal that we're developing, consider what you'd do if

- an employee leaves the company to join that pesky competitor, Martian Resort and Luxury Spa.
- an employee joins the resort as a new Mechanical Crew member.
- an employee is promoted from Crew Supervisor to Department Head, and needs the requisite permissions within Liferay DXP.
- you need to organize the users by department.
- a new department is added to the Lunar Resort and the employees need their own internal website.
- an employee gets married, and their name changes.

The user tasks listed above are all resolved in the Users and Organizations section of the Control Panel.

Figure 27.2: The Users section of the Control Panel. This screenshot shows a portal that doesn't have the EE Audit plugin installed.

What are Users?

In case there's any confusion over the term, a User in Liferay is an entity that can sign into the portal and do something. Generally a User has more privileges, called Permissions, than a Guest of your site, who does not sign in. Users are assigned a Role (the Power User Role by default), and a Role is what defines the user's privileges.

Understanding users is pretty straightforward. So how do you, the administrator, add users to the portal?

Adding Users

As the Lunar Resort Administrative user, you can add Users to the portal.

1. From the Product Menu, click *Control Panel* → *Users* → *Users and Organizations*.
2. In the Users tab, click the Add button ( ).

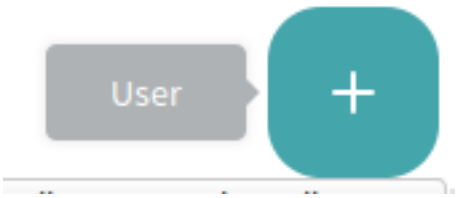


Figure 27.3: Add Users from the Users and Organizations section of the Control Panel.

3. Fill out the Add User form and click *Save*. At a minimum, provide a Screen Name, First Name, Last Name, and Email Address for the User.

   Below the form, you can continue configuring the user by clicking either Organizations or Personal Site Template.

Once you save the form, the User is added to the portal.

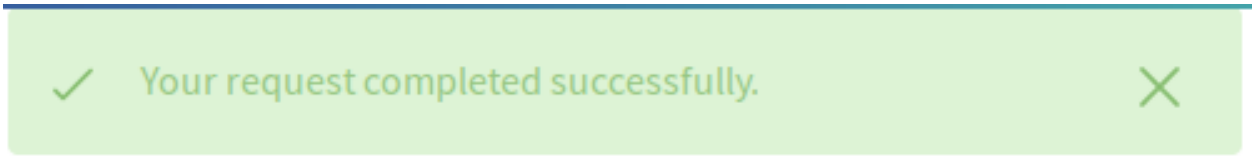


Figure 27.4: At a minimum, enter a screen name, email address, and first name to create a new user account.

After you submit the form, the page reloads with a success message. An expanded form appears for editing the user's account (see all the new categories listed below the form?). This lets you fill out a lot more information about the user. You don't have to fill anything else out right now. Just note that when the user account was created, a password was automatically generated. If Liferay was correctly installed and a mail server was set up (see here), an email message with the user's new password was sent to the user's email address. This, of course, requires that Liferay can properly communicate with your SMTP mail server.

If you haven't yet set up a mail server, use this page to change the default password for your user account to something you can remember. Clicking on the *Password* section heading below the form opens the Password section of the Add User form. Enter the new password twice and click *Save*.

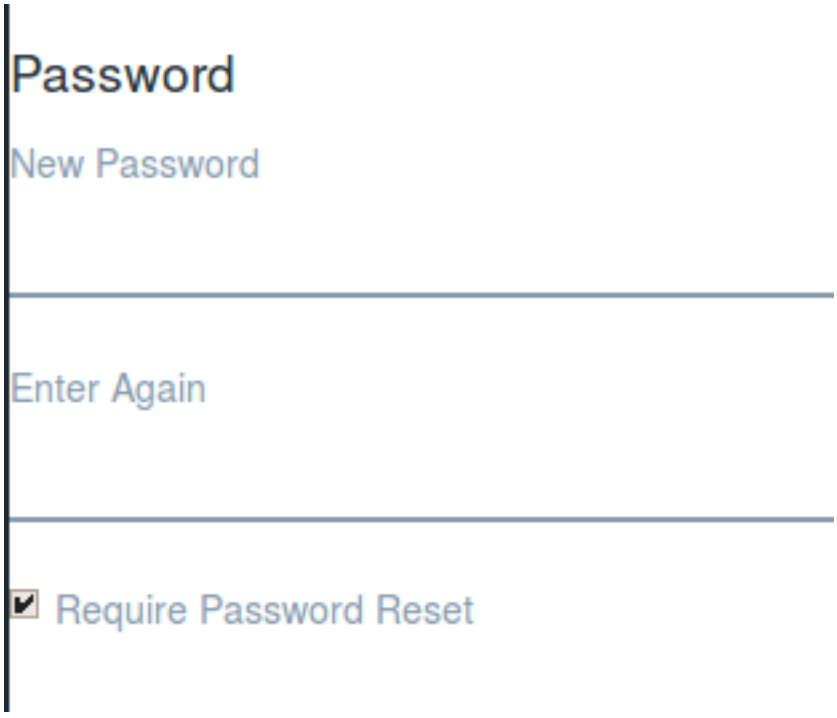


Figure 27.5: Enter the password twice to manually set the password for a user. If the Password Policy you're using is configured to allow it, select whether to require the user to reset their password the first time they sign in to the portal.

Note: See the *Require Password Reset* checkbox at the bottom of the Password form? The default password policy is set so that even you, the administrator, cannot deselect it. As the administrator, however, you do have the ability to modify the default password policy so that this box becomes usable. Navigate to Password Policies in the Control Panel, open the Actions menu for the Default Password Policy, open the *Password Changes* section, and toggle the *Change Required* checkbox to *NO* and save. Now you can decide whether each user you add will need to reset their password.

See the article on Password Policies for more information on editing the default policy or creating your own.

Adding an Administrative User What if you are just setting up the Lunar Resort portal for the first time, and you're using the default administrator account, the account of one of those famous Liferay Administrators, *Test Test* or her cousin, *Joe Bloggs*? Since you're the administrator of the Lunar Resort portal, you want to set up your own administrator account for the portal. Use the steps above to add a user with your information, then give your user account the same administrative rights as the default administrator's account, click the *Roles* link below the form. This section shows the roles to which your account is currently assigned. At the moment, no roles are listed.

Note: You can also customize the default roles a new user receives via *Default User Associations*. This is covered in the article on Portal Settings.

Assign the Administrator role to your user account. On the form for editing a user, after having clicked on the *Roles* section below the form, click the *Select* button above Inherited Regular Roles. A dialog box pops up with a list of all the regular (portal-scoped) roles in the portal. Select the Administrator role from the list. The dialog box disappears and the role is added to the list of roles associated with your account. Don't forget to click the *Save* button which is at the bottom of the page, below all the sections. You are now a portal administrator. Log out of the portal and then log back in with your own user account.

In production, you should always delete or disable the default administrator account to secure your portal.

Next, let's look at some other aspects of user management.

Editing Users

If you click on *Users and Organizations* in the Control Panel, you'll see your own user's account in the list of users, along with any others. If you want to change something about a particular user, you can click the *Actions* button (the vertical ellipsis icon) next to that user.

- Choosing *Edit* takes you back to the Edit User page where you can modify any aspect of the user account including the screen name, email address, first name, last name, site and organization memberships, roles, etc.
- Choosing *Permissions* allows you to define which roles have permissions to edit the user.
- Choosing *Manage Pages* allows you to configure the personal pages of a user.

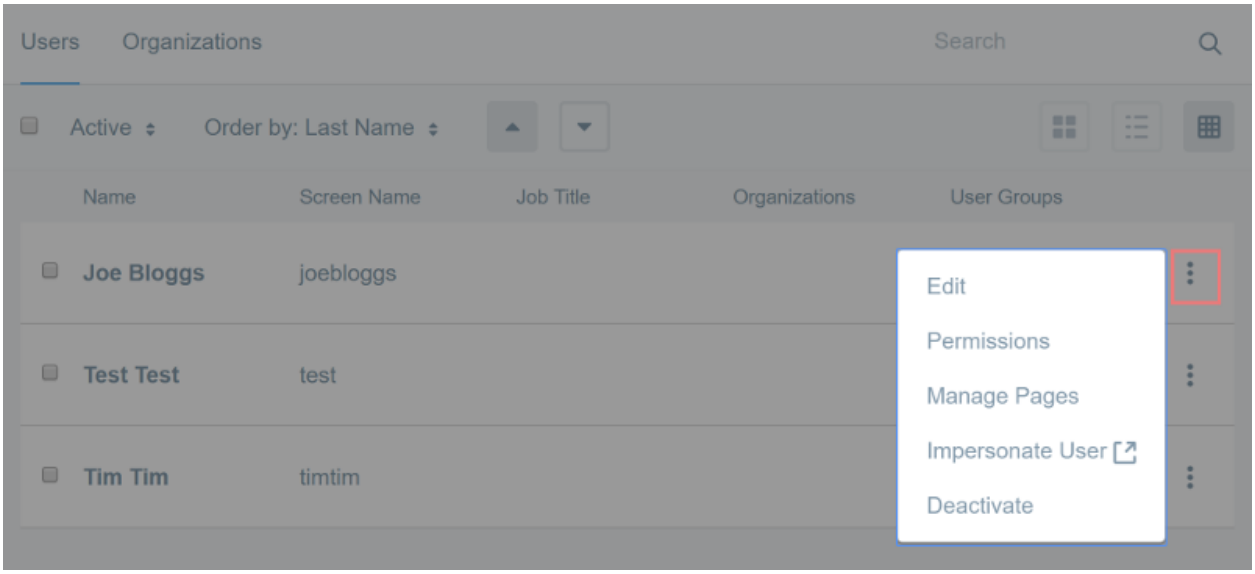


Figure 27.6: Open the Actions menu next to a listed user to update its configuration.

- Choosing *Impersonate User* opens another browser window which allows you to browse the site as if you were the user. This allows you to test your user management on a user to make sure you're achieving the desired behavior, without having to repeatedly log out of your administrator account and into the user's account.
- Choosing *Deactivate* deactivates the user's account. The user will still be in your database, along with all the rest of your users, but their account will be deactivated, so that they cannot sign in to the portal. If you have any deactivated users, you'll have the option to view active portal users or inactive portal users. If all the users are active, this filtering option will not appear.

Note that most users can't perform most of the above actions. In fact, most users won't have access to the Control Panel at all. You can perform all of the above functions because you have administrative access.

User Images

Users in Liferay DXP have avatars. Administrative users can upload images in the Edit User form.

If no image is explicitly uploaded for a user's avatar, a default image is assigned, using the initials of the user (First Name then Last Name) over a random color.

For many locales, the default approach for generating user images is perfectly suitable. For some locales, though, the default user image is misleading or confusing. For example, there are locales where the order of the initials is reversed (Last Name then First Name), and there are locales where the use of one character of the first name and one character of the last name produces meaningless results to the reader (especially languages not based on the Latin alphabet). If that's true for your locale, disable the inclusion of users' initials in the default avatars. Just enter

```
users.image.default.use.initials=false
```

in a `portal-ext.properties` file placed in your Liferay Home folder. Once you restart Liferay DXP, the new default images will be used to generate user avatars.

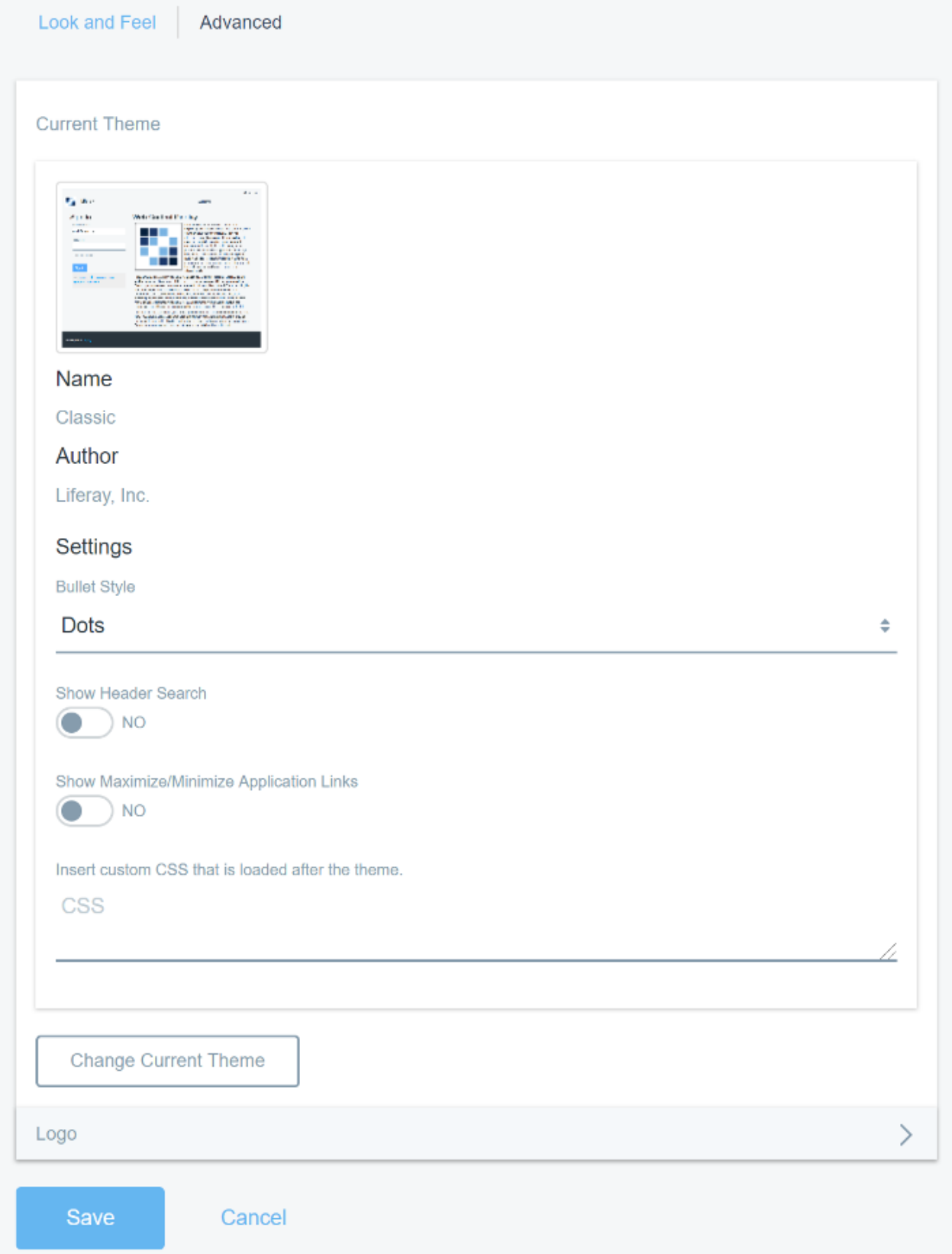


Figure 27.7: You can manage a user's personal pages through the user's Actions menu.

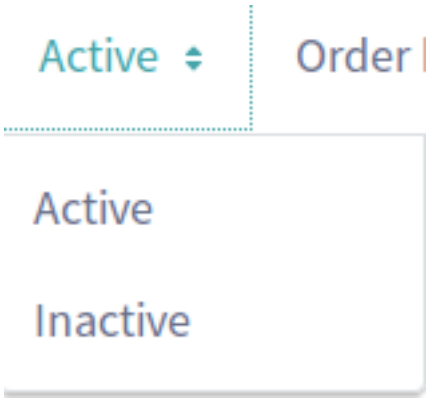


Figure 27.8: You can choose whether to view active or inactive (deactivated) portal users in the users list found at *Product Menu → Control Panel → Users → Users and Organizations*.

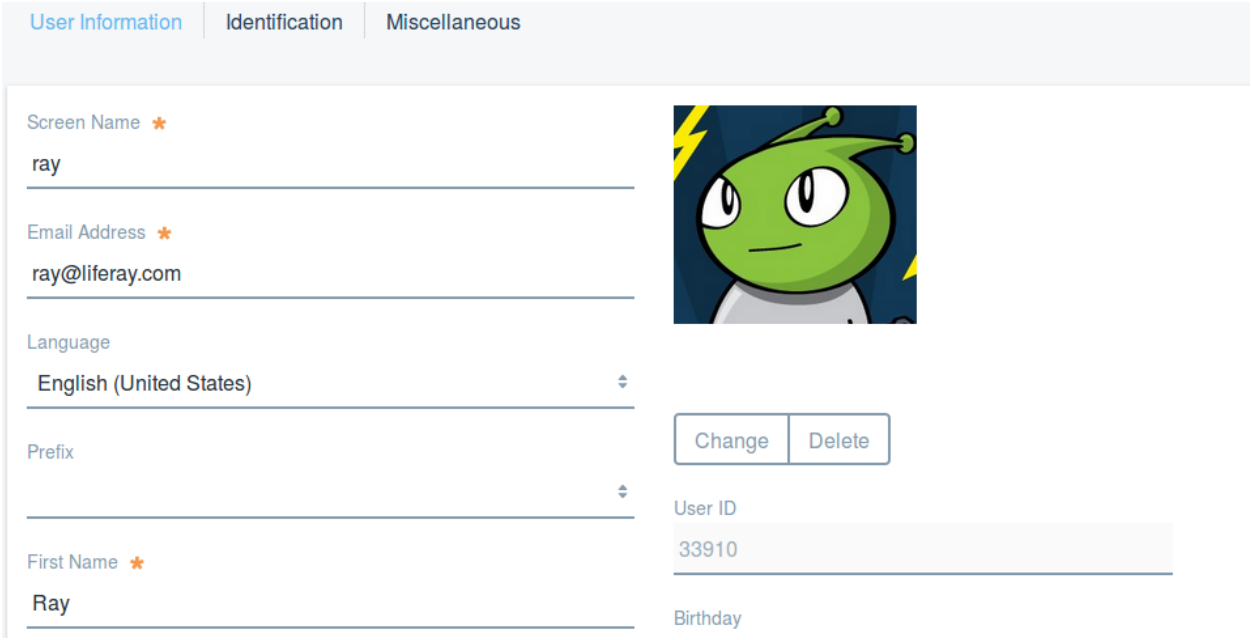


Figure 27.9: Upload images for user avatars in the Edit User form.

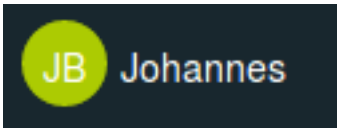


Figure 27.10: If Johannes Bach was a user in your Liferay DXP instance, his default avatar might look like this.

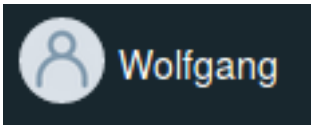


Figure 27.11: Wolfgang Amadeus Mozart's default avatar, after disabling the use of user initials.

Learn how to take advantage of more localization strategies by leveraging language settings. Next, learn about collecting users in organizations.

27.2 Adding and Managing Organizations

For some enterprises, particularly those that are organized in a hierarchical structure, using organizations as a user management tool can make a lot of sense, and can make the delegation of user management responsibilities much easier than it would be otherwise. To understand what a hierarchical structure looks like, consider Major League Baseball (MLB). There's the main MLB organization, The American and National Leagues, and each League's Divisions (for example, the American League's East Division). Finally, the Individual Teams are under the Divisions. This structure could be matched in the MLB portal using organizations.

As with users, organizations are managed in the Control Panel. The steps for creating and managing organizations can be found after the next section, but if you're not sure what an organization really is or whether you need organizations in your portal, read this next section to help you wrap your head around it.

What are Organizations?

An Organization is a way to group Liferay Users hierarchically. For example, your company's departments (Human Resources, for example) could be an organization in Liferay. Often times, organizations have their own site.

Many simple portal designs don't use organizations at all; they only use sites (see the sections on Creating Sites for more information on sites). The main purpose of organizations is to allow for distributed user management. Using organizations, portal administrators can delegate some user management responsibilities to organization administrators. If you don't anticipate needing to delegate user management responsibilities, your portal design probably doesn't need to include organizations. In order to decide whether or not your portal design should include organization, think about your portal's function. A simple photo-sharing web site could be powered by sites only. On the other hand, organizations are useful for corporations or educational institutions since their users can easily be placed into a hierarchical structure. Don't think that organizations are only for large enterprises, though. Any group hierarchy, from large government agencies all the way down to small clubs, can be modeled with organizations. Also, don't think that you must decide between an organization based structure or a site based structure for assembling your portal's users. Users can belong both to organizations and to independent sites. For example, a corporation or educational institution could create a social networking site open to all portal users, even ones from separate organizations.

To illustrate what an Organization is, consider a potential organization of the Lunar Resort's intranet. The company hierarchy has three tiers: The Lunar Resort, its Departments, and divisions within each department.

- Lunar Resort–The top-level Organization.
  - Physical Plant Department–Department of users that keep the place running.
    - Grounds Crew–Users that maintain the grounds.
    - Janitorial Crew–Users who keep the resort clean.
    - Mechanical Crew–Users who fix stuff, like Lunar rovers.
  - Recreation Department–A department that makes sure much fun is had by guests of the Lunar Resort.
    - Golf Instructors–Teach guests how to golf on the moon.
    - Rover Race Instructors–Teach guests how to drive the Lunar Rovers.

    * Lunar Sherpas–Lead guests on moon hikes.
- Sales Department–A department of users who sell things to Lunar Resort Guests.
    * Up-sale Group–Make sure Guests know how easy it is to improve their stay by spending more money.
    * Souvenir and Memorabilia Group–Peddle souvenirs to Lunar Resort Guests.
    * Retail Group–Maintain the Lunar Resort store, which contains basic necessities, since guests are coming all the way from Earth.

Each department is a sub-organization of the resort, and each further division is a sub-organization of the department.

Note: It's easy to confuse User Groups (covered in a separate article) and Organizations. Both are used to group users, but User Groups are usually an ad hoc collection of users, organized for a specific function in the portal. In the Lunar Resort, if you wanted a group of bloggers, for example, it wouldn't make sense to assign the Sales Department the role of blogging (again, see the article on roles if you're not sure what they are). The Sales Department users would be able to blog whenever a new t-shirt design became available in the Lunar Resort store, but they probably wouldn't be as diligent about announcing the new Rover Racing schedule. Instead, creating a user group containing one individual from each department who is responsible for blogging would make more sense. Read the article on User Groups to learn more about how to use them in your portal.

Whenever you have a collection of users that fit into a hierarchical structure, you can use organizations to model those users. Organization administrators can manage all the users in their organization *and* in any sub-organization. Referring to the hierarchy above, for example, an organization administrator of the Lunar Resort could manage any users belonging to the resort itself, to any of the departments, or to any of a department's subdivisions. An Organization Administrator of the Physical Plant Department can manage any users belonging to the Physical Plant Department itself, or to the Grounds Crew, the Janitorial Crew, or the Mechanical Crew. However, an administrator of the Physical Plant Department can't manage users belonging to the Recreation Department or users in the Retail Group organization.

Organizations and sub-organization hierarchies can be created to unlimited levels. Users can be members of one or many organizations. The rights of an Organization Administrator apply both to his/her organization and to any child organizations. Members of child organizations are implicit members of their parent organizations. This means, for example, that members of child organizations can access the private pages of their parent organizations. This behavior can be customized in your portal's `portal-ext.properties` configuration file. There's an `Organizations` section of the portal.properties file where the properties specific to organizations are listed.

Since organizations are designed for distributed user administration, Organization Administrators have an entirely different set of privileges than Site Administrators. Site Administrators are responsible for the pages, portlets, and content of their site. They are also responsible for managing the membership of their site. To this end, they can set the membership type to Open, Restricted, or Private. They can also add users to or remove users from their site but cannot manage the users themselves. If an organization has a site attached to it, the Organization Administrator has the same administrative ability as a Site Administrator when it comes to managing the site's content, but the site membership management is different. An organization site's members will simply be the members of the organization. Organization Administrators have more user management permissions than Site Administrators; they can edit users belonging to their organization or any sub-organization. They cannot add existing portal users to their organization, but they can create new users within their organization. Only portal administrators can add existing users to an organization.

Organization Administrators can't access the Control Panel by default, but they don't need to. Under their *My Account* menu, in the Control Menu, organization administrators can click on the *My Organizations* link to gain access to any organizations they manage.

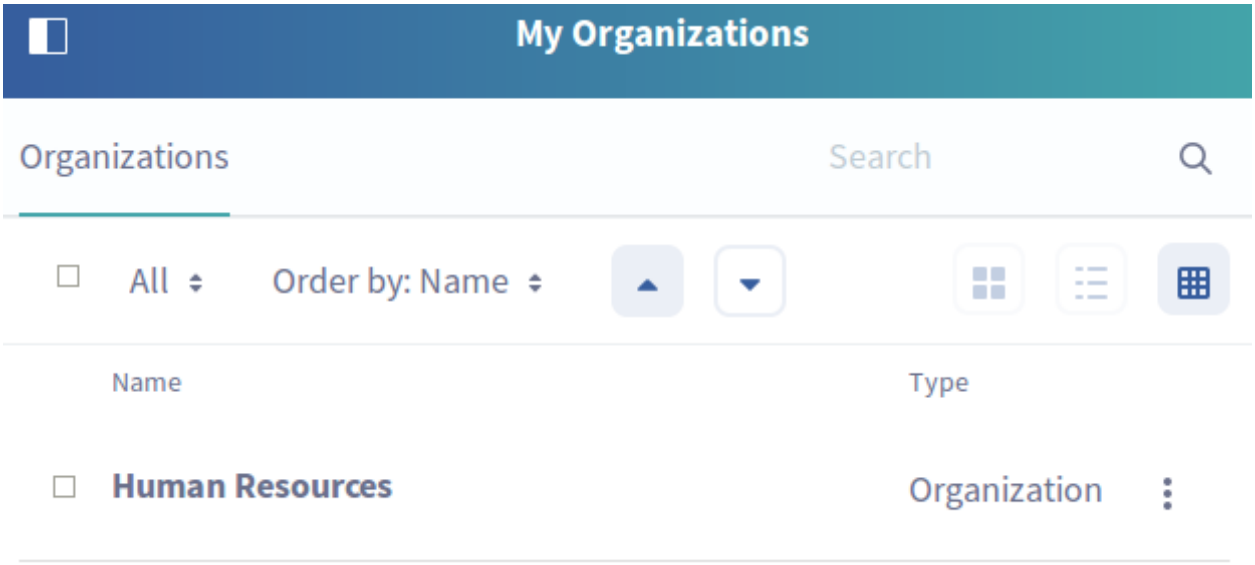


Figure 27.12: The My Organizations application lets Organization Administrators manage their organizations in their personal site.

A huge time-saving benefit of including organizations into your portal design is that organization administrators can assign organization-scoped roles to members of the organization. For example, consider an IT Security group in a corporate setting. You could have a sub-organization of your IT organization that handles security for all of the applications company-wide. If you grant the IT Security organization the portal administrator role, all the members of the organization would have administrative access to the entire portal. Suppose further that a user in this organization was later hired by the Human Resources department. The simple act of removing the user from the IT Security organization also removes the user's administrative privileges, since the privilege came from the IT Security organization's role. By adding the user to the HR organization, any roles the HR organization has (such as access to a benefits system in the portal) are transferred to the user. In this manner, you can design your portal to correspond with your existing organization chart and users' permissions are granted according to their positions in the chart.

Of course, this is only one way to set up your portal. If you have more complex requirements for permissions within an organization, you can create custom organization-scoped roles to assemble the permissions you wish to grant to particular users. Alternatively, you could consider attaching a site to your organization and using site teams to assemble the sets of permissions (see below). See the Roles and Permissions article for more detail.

Does your organization need to have its own site? Many organizations don't, but since some do, Liferay allows sites to be attached to organizations. If an organization has an attached site, the organization's administrators are treated as the site administrators of the attached site. This means that they can manage the pages, portlets, and content of the site as well as the users of the organization. Members of an organization with an attached site are treated as members of the organization's site. This means that they can access the private pages of the organization's site, along with any portlets or content there. The capability of attaching sites to organizations allows portal administrators to use organizations to facilitate distributed portal administration, not just distributed user administration.

That's a lot of information on organizations. Next, learn how to create and manage organizations.

Adding Organizations

Now add an organization to the portal (perhaps start by adding the *Physical Plant Department* organization to the Lunar Resort):

1. Open the Control Menu and navigate to *Control Panel → Users → Users and Organizations*.

2. Select the *Organizations* tab and click the *Add* button to add a new organization. To attach a site when you create an organization, click on the *Organization Site* section heading at the bottom and check the *Create Site* box. If you don't know right now if your organization needs a site, that's fine. You can always add one later.

```
![ Use the Add Organization form to create a new Organization for the Lunar Resort.](./images/organizations-add-organization.png)

- Enter a Name for the organization.
- Select an organization in the system to be the direct parent of the
    organization you are creating. Click the *Remove* button to remove the
    currently configured parent.
- Click *Save* when finished filling out the Add Organization form.
```

As when creating a new user, once you submit the form a success message appears and you have access to a new form which lets you enter additional information about the organization. The form is organized under three tabs:

- *Organization Information:* contains the basic information that you provided when you created the Organization, along with a few additional settings for an avatar image and Categorization.

- *Identification:* specifies the contact information for the organization. Organizations can have multiple email addresses, postal addresses, web sites, and phone numbers associated with them. The Services section form can be used to indicate the operating hours of the organization, if any.

- *Miscellaneous:* contains comments about the organization, reminder queries for the users of the organization, and any custom fields for the organization.

Tip: After creating an organization, you should assign the desired user to the Organization Owner Role. The Organization Owner can do everything that an Organization Administrator can. In addition to their full administrative rights within the organization, they can do these things:

- Appoint other users to be Organization Administrators
- Appoint other users to be Organization Owners
- Remove the memberships of other Organization Administrators or Owners

Organization Administrators can't make these role assignments and can't manage the memberships of other Organization Administrators or Owners.

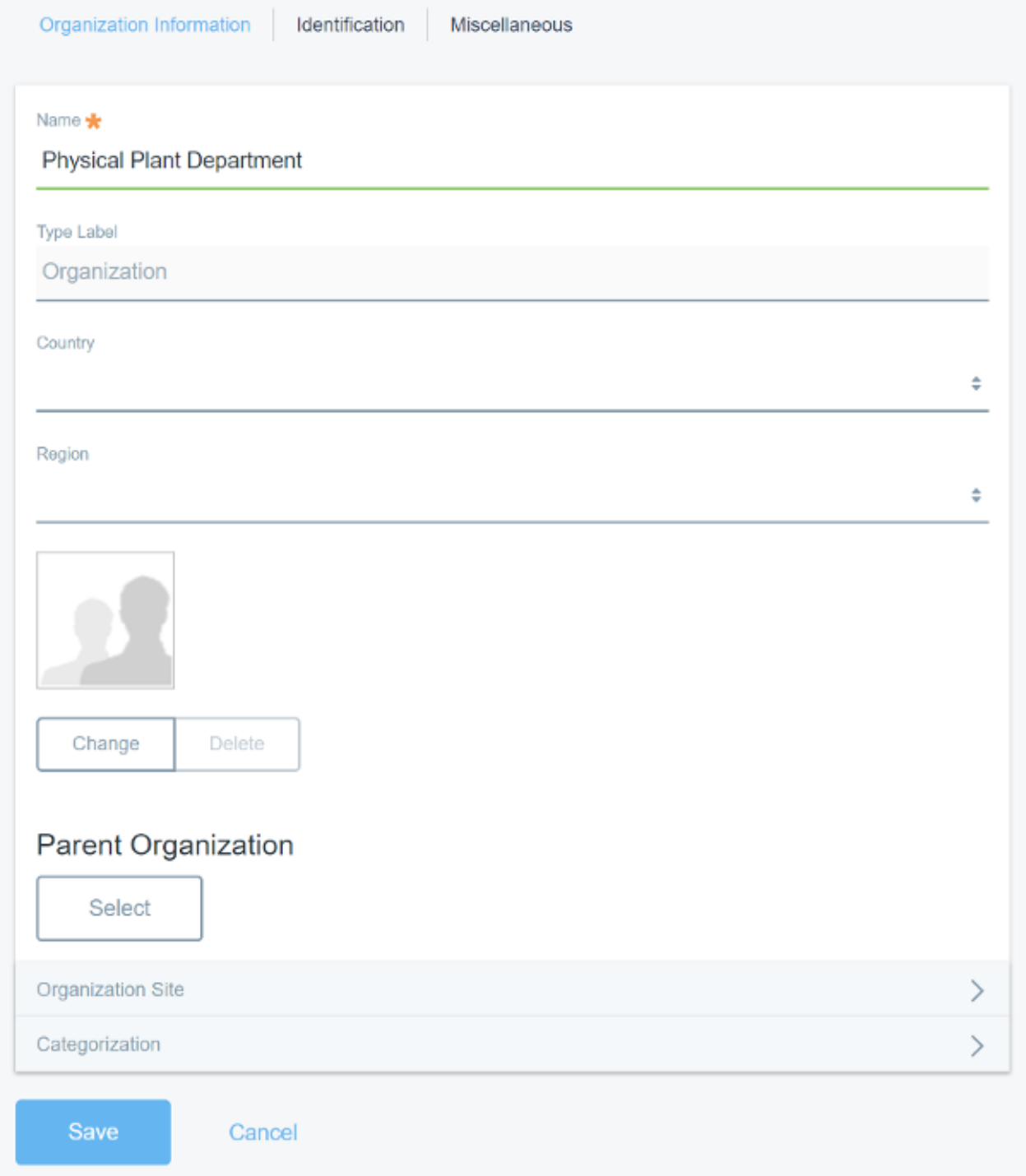


Figure 27.13: Once the Add Organization form is submitted you can provide additional information about the organization.

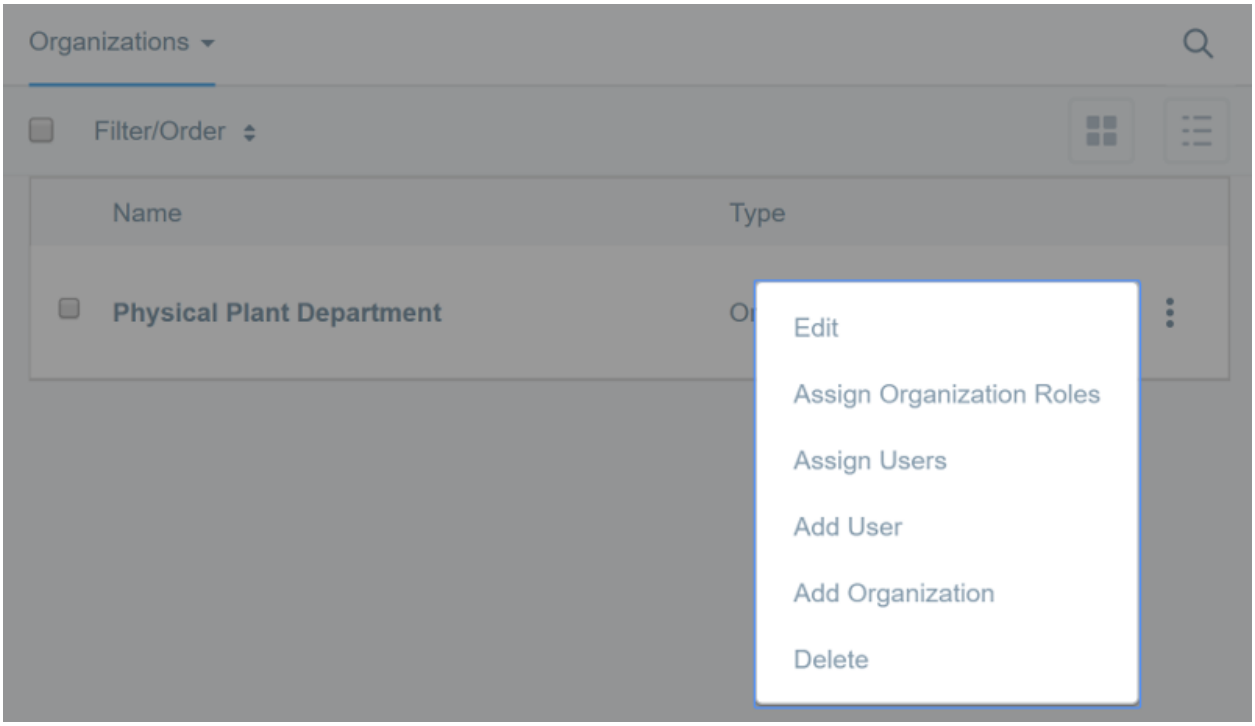


Figure 27.14: Open the Actions menu next to an organization to update its configuration.

Editing Organizations

To edit an organization go to the Users and Organizations section of the Control Panel and click the *Organizations* tab. You should see any active portal organizations listed. Click the *Actions* button (the vertical ellipsis icon) next to an organization. This shows a list of actions you can perform on this organization.

- *Edit* lets you specify details about the organization, including addresses, phone numbers, email addresses and websites.
- *Manage Site* lets you create and manage the public and private pages of the organization's site. This only appears for organizations that have attached sites.
- *Assign Organization Roles* lets you assign organization-scoped roles to users. By default, Organizations are created with three roles: Organization Administrator, Organization User and Organization Owner. You can assign one or more of these roles to users in the organization. All members of the organization automatically get the Organization User role so this role is hidden when you click Assign Organization Roles.
- *Assign Users* lets you search and select users in the portal to be assigned to this organization as members.
- *Add User* adds a new user in the portal and assigns the user as a member of this organization.
- *Add Organization* lets you add a child organization to this organization. This is how you create hierarchies of organizations with parent-child relationships.
- *Delete* removes this organization from the portal. Make sure the organization has no users in it first.

If you click the *View* button at the top of the Users and Organizations page and select *View Hierarchy* you can view both a list of users who are members of this organization and a list of all the sub-organizations of this organization.

Organization Types

By default, Liferay DXP only includes the *Organization* type. Configure additional organization types using `portal.properties`. There are two main reasons an enterprise wants to configure organization types:

1. Organizations usually correlate to real-life hierarchical structures. Calling them by their real names is helpful for administrators and users. In the Major League Baseball (MLB) example, *League*, *Division*, and *Team* organization types are useful.
2. Enforce control over which organizations can be top level organizations and the type of sub-organization allowed for each parent organization type. For example, MLB would not allow Division organization types to be sub-organizations of Team organizations.

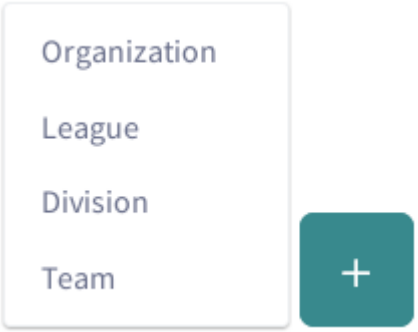


Figure 27.15: Make additional organization types available in the Control Panel by adding them to the `organizations.types` portal property.

Check out the portal properties that configure the default *Organization* type on docs.liferay.com. To add another organization type called *League*, add this to `portal-ext.properties`:

```
organizations.types=organization,League
organizations.rootable[League]=true
organizations.children.types[League]=Division,Team
organizations.country.enabled[League]=true
organizations.country.required[League]=false
```

So what do all those properties do?

- `organizations.types=organization,League`: adds League to the list of organization types that appear in the Add Organization menu.
- `organizations.rootable[League]=true`: enables Leagues as a top level organization. Limit League to sub-organization status by excluding this property.
- `organizations.children.types[League]=Division`: specifies Division as the only allowable sub-organization type for the League parent type.
- `organizations.country.enabled[League]=true`: enables the Country selection list field on the form for adding and editing League types.
- `organizations.country.required[League]=false`: specifies that the *Country* field is not required when adding a League.

Once you configure additional organization types in `portal-ext.properties`, restart the server and you'll see your new type(s) in the Organizations section of the Control Panel.

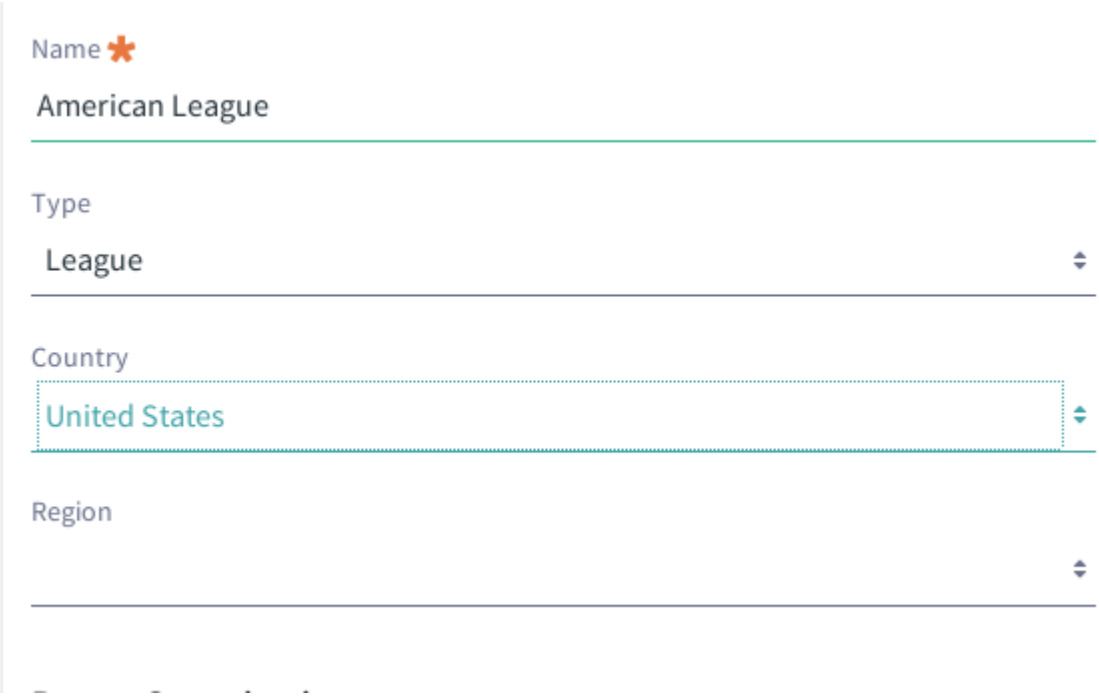


Figure 27.16: Add the Country select list field to the Add Organization form with the `organizations.country[my-org-type].enabled` property.

Users can join or be assigned to sites when they share a common interest. Users can be assigned to organizations when they fit into a hierarchical structure. Users groups provide a more ad hoc way to group users than sites and organizations. Let's look at them next.

27.3 User Groups

A User Group is a group of users. Were you hoping for a more complicated, or enterpris-ey, definition? Try this: A user group is a box. What's inside the box? Users. Is that definition still lacking? A user group collects users that don't fit together outside of the specific function that the user group was formed to fulfill. That's really all it is. Of course, this is Liferay, so there's a ton of flexibility built in to user groups, and they're integrated with Roles, Sites, Site Templates, Permissions, and, of course Users. Because of that, there are a lot of different places where you can interact with user groups, and there's more than one use case they can fulfill. In this article you'll learn how to work with user groups to serve the most common use cases.

User groups allow portal administrators to create groups of users that traverse hierarchical boundaries. For example, consider The Lunar Resort. It's hierarchically divided into the Physical Plant Department, the Recreation Department, and the Sales Department. The Lunar Resort's department-driven structure is modeled using organizations. If there are functions needed in The Lunar Resort public or private sites that can't be divided up according to the existing organizational hierarchy, you might need user groups. For example, it might make sense to create a user groups that includes the Human Resources representatives of each department. User Groups are most often used to achieve one of the following goals:

- Collect permissions. With User Groups, you can assign roles and permissions to a group of users that don't share an Organization. For example, in a University portal, a user group could be created to

group all teachers independently of their organizations to make it easier to assign one or several roles at once to all the teachers. In The Lunar Resort, we're going to use this approach to give members of several organizations the *List Creator* role.

- Manage site membership. Using the previous example, all teachers could be members of the sites *University Employees* and *Students and Teachers Collaboration Site* by adding the *Teachers* user group as a member. In the Lunar Resort, you can add all the List Creators to the public facing site of The Lunar Resort.
- Manage user personal pages. Provide predefined public or private pages to the users who belong to the user group. For example, the *Teachers* user group could be created to ensure the home page on all teachers' personal sites has the same layout and applications. In The Lunar Resort, you'll add a site template that adds an Event Coordination page to each List Creator.

At the Lunar Resort, certain employees need the ability to create simple form applications using Liferay's Dynamic Data Lists. On the harsh lunar landscape, it's important to have a proper idea how many guests are coming to the next Lunar Hike event. Many of the list creators will be users from the Recreation Department, since many of the activities that guests should know about are managed by that department. So why not just give all the users of the Recreation Department the proper permissions for managing the Dynamic Data Display portlets on the Activities Sign-Up page, and be done with it? Sometimes other departments will need to add an event (for example, the kitchen staff is preparing a Lunar Luau and needs a guest list). Also, you might not want all the users of the Recreation Department to be a List Creator. Remember that time Wilbert Donahue (he's a Lunar Rover Race Instructor, part of the Recreation Department) was granted blogging permissions, and he used the opportunity to write all about his favorite band, *ApeFLogger*? You don't want Wilbert to be a List Creator, right?

Creating a User Group

User Groups are administered in the Control Panel, under *Users* → *User Groups*. Start by creating a user group called *List Creators*. Later, you'll also create a new site template for a user group site that user group members can access to coordinate event scheduling, and you'll use the group to assign all its users to The Lunar Resort public site. First create the user group:

1. Navigate to the Control Panel by opening the Product Menu (click) and clicking *Control Panel*.
2. Click on the *User Groups* link under the *Users* menu and then click on the *Add* button.

   Name is the only required field, but you should enter a description as well.
3. Click *Save* and you will be redirected back to the *User Groups* page of the Control Panel.

Right now, the user group is just an empty box. The user group has no sites or roles to give it any functionality. No users are assigned to the user group, so it isn't collecting users yet. That comes next.

Assigning Members to a User Group

To add existing users to a user group:

1. Navigate to the User Groups page of the Control Panel and click *Actions* ( ) → *Assign Members* next to the List Creators group.

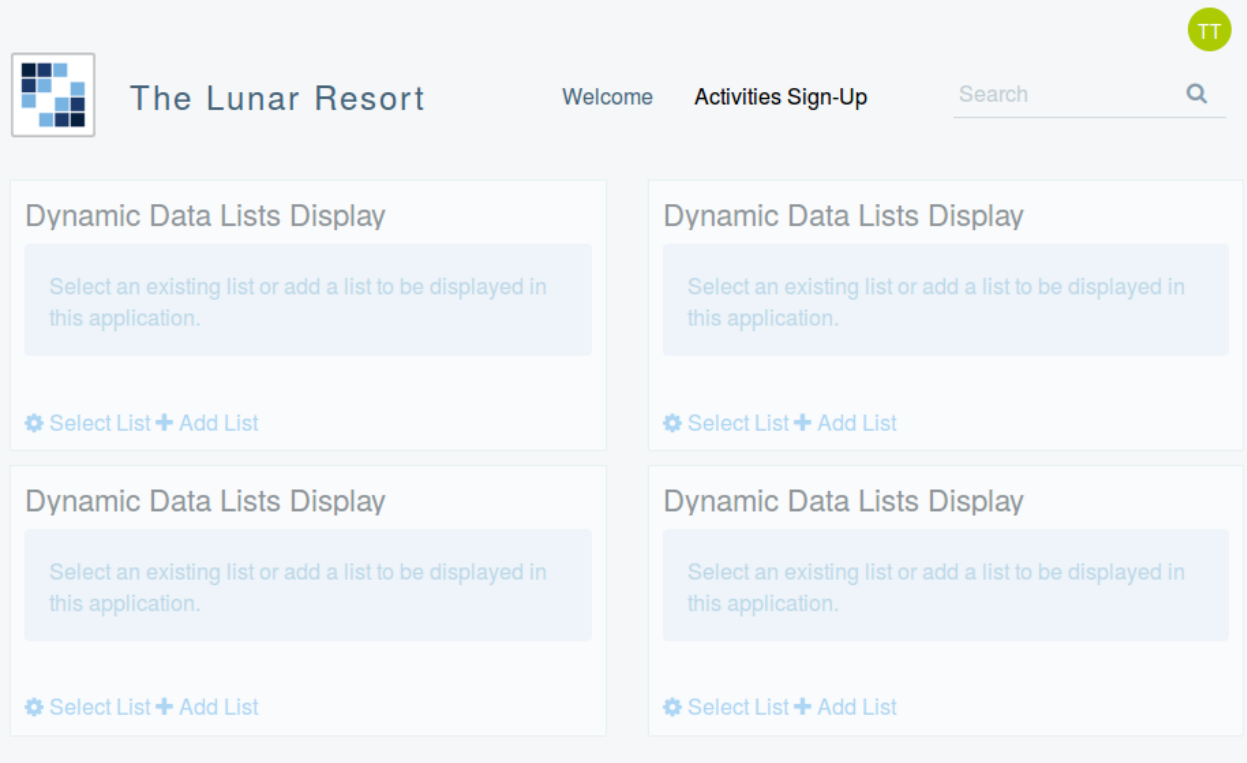


Figure 27.17: The List Creators group is needed to manage the sign-up lists on the Activities Sign-Up page of The Lunar Resort.

2. Click the *Add Users* button to see a list of users that can be assigned to the group.

3. From this list, select one or more users and click *Add*.

Once you have a user group with users in it, what are you going to do with it? The List Creators group is destined to get some specific permissions in the Lunar Resort, but you can use user groups for site membership management or personal page management. These scenarios are all covered in this article.

Next, add the users in the List Creators group to the Lunar Resort site.

User Groups and Site Membership

After a user group is created, you can add all its users at once as members of a site in one step from the *Site Memberships* UI. In this example you'll see how the newly created List Creators user group can be added to The Lunar Resort site. This makes managing site membership easy. Just make sure you have the correct site selected in the Product Menu!

Associate the user group with a site:

1. From the Product Menu, select the site you want to work in (The Lunar Resort) and open the site administration drop-down.

2. Click *Members → Site Memberships*.

3. Click on the *User Groups* tab, click the Add button ( ) and select the List Creators user group.

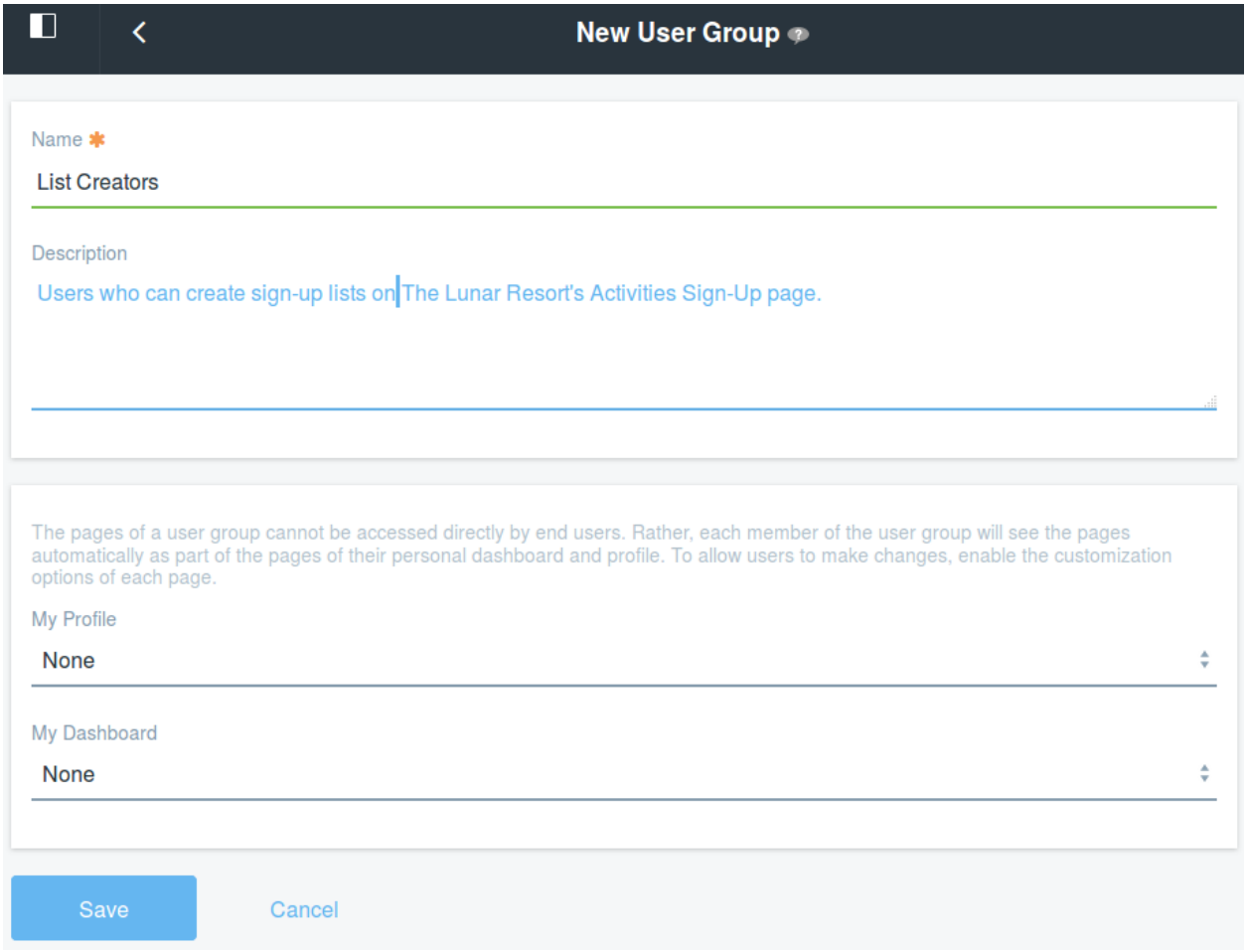


Figure 27.18: Provide a name and a description for your new User Group.

Now members of the List Creators user group are site members of The Lunar Resort site. Managing site membership with user groups is easy. User groups can also have their own sites.

User Group Sites

Each Liferay user has a personal site, consisting of public (Profile) and private (Dashboard) pages. A user group site determines the base pages of the personal sites of all the user group members. If the user group site pages are added to a user's Profile pages, then the user group site is a public site, accessible to anyone with the URL (`http://www.[sitename].com/web/[username]`). If the site pages are added to the user's Dashboard pages, then the site is a private site. A mixed approach can also be used, where both private and public pages are added for the user group site. If a user belongs to multiple user groups, all the pages from those user group sites are made part of his personal site.

When you created the List Creators user group, did you notice the section of the New User Group form that allowed you to select a site template from two drop down menus, *My Profile* and *My Dashboard*? That's where you add user group sites from a site template. You can add a user group site while adding the user group initially, or add one later if you decide you need to add one. You can also add pages to the user group site manually, if you don't want to use a site template. But that's done later, from the Actions menu of an

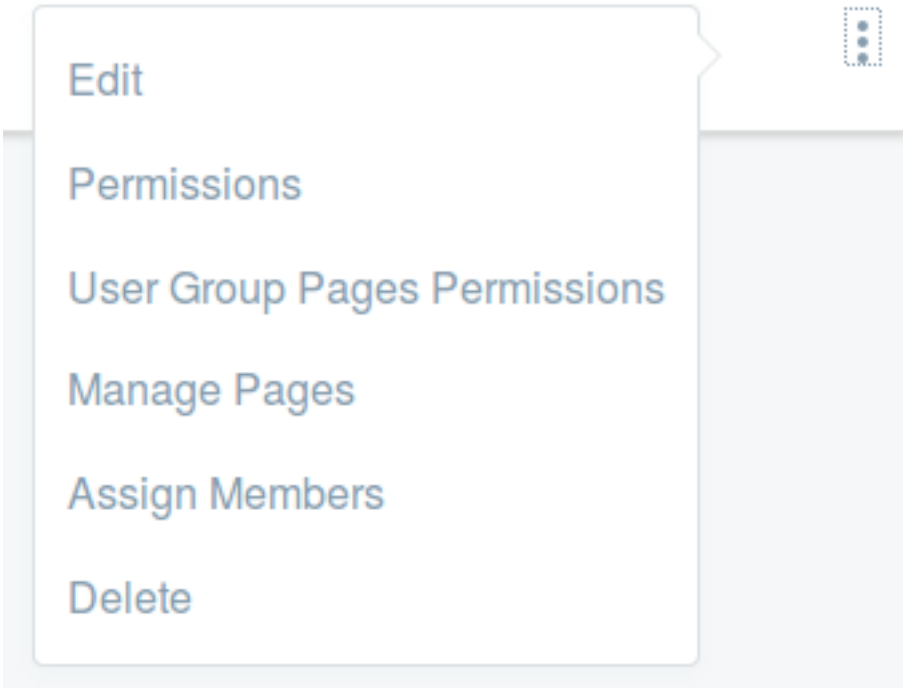


Figure 27.19: To add users to the User Group, select *Assign Members* from its Action menu.

Figure 27.20: Add users from the list of available users.

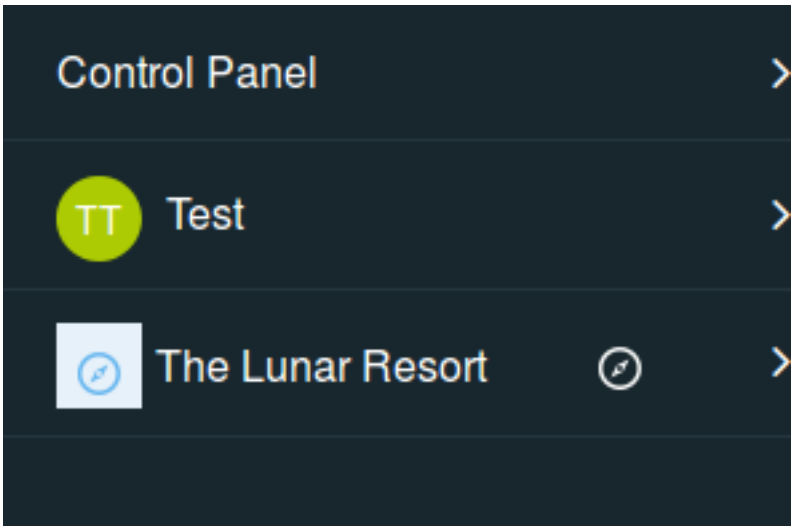


Figure 27.21: Select the site you want to manage using the site selector.

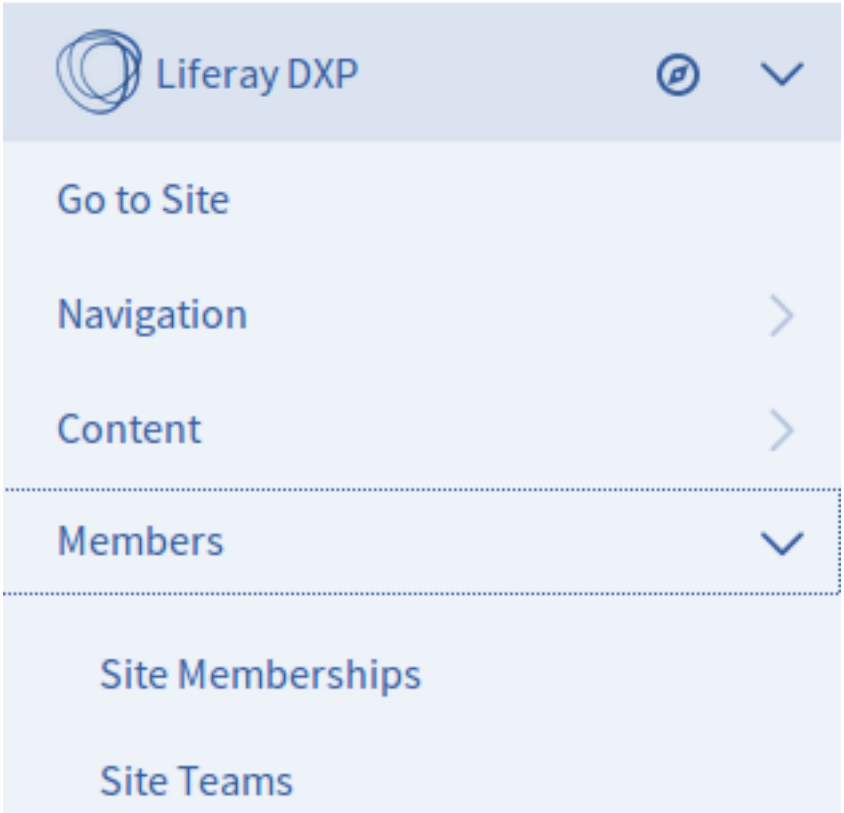


Figure 27.22: Manage site memberships from the Product Menu.

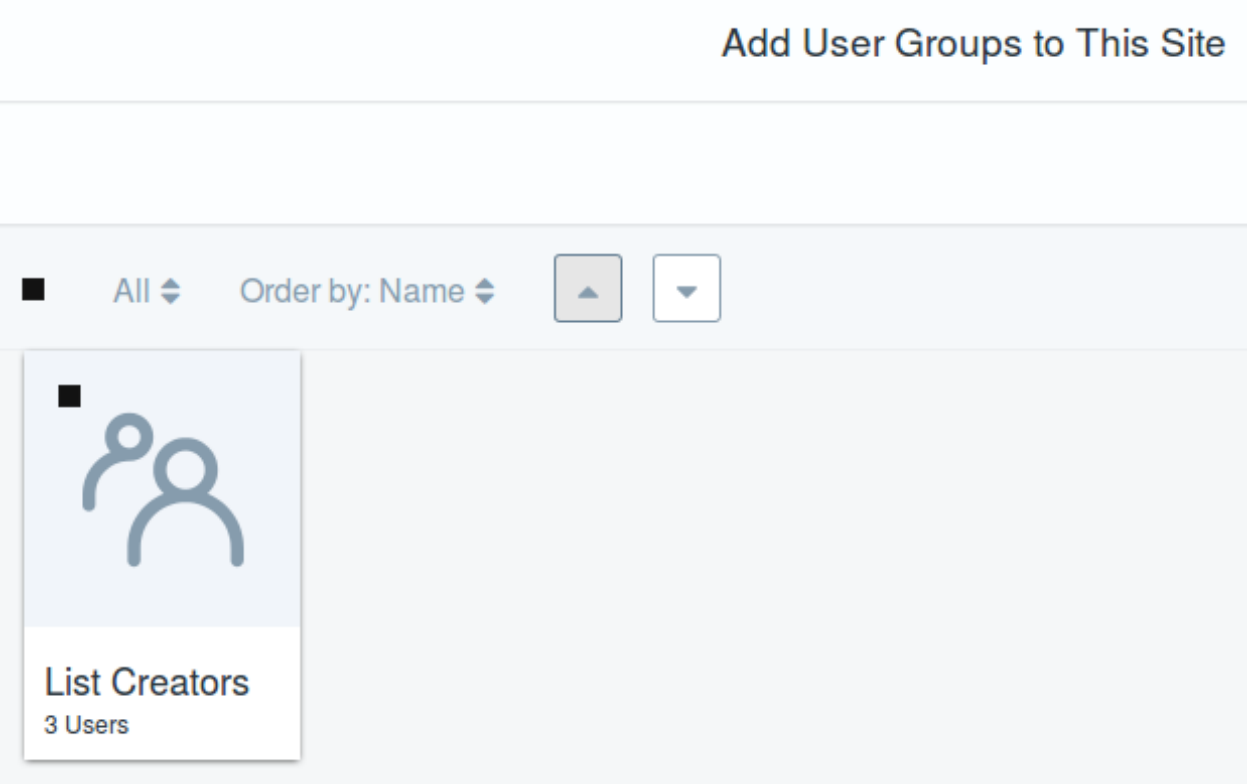


Figure 27.23: User groups let you assign many users to a site at once.

already-created user group.

In The Lunar Resort, the List Creators must have access to the private *Event Coordination* page, with a Calendar application where events are scheduled that the user group members will create lists for, and a Message Boards portlet to discuss and resolve scheduling conflicts. Using site templates along with the Calendar and Message Boards applications, you can easily do this. First you need to create a site template for use in the user group site and deploy the applications to the site pages.

To create a new site template, navigate to *Sites* → *Site Templates*. Add a site template called Event Coordination and save it. Click the *Event Coordination* template link and rename the default Home page to Event Coordination. Now there's a site template, but the page is currently empty. Go to the Event Coordination page, and add a Message Boards and a Calendar application. Read the documentation on site templates if you need further explanation.

Once you have a site template set up, use it to create the user group site.

1. Navigate to the *Users* → *User Groups* section of the Control Panel.

2. To edit the List Creators user group, click the name of the group or the Actions button (⋮), then *Edit*, for the user group.

3. Since the pages should be private pages for the users of the site, open the drop-down menu under *My Dashboard* and select the new site template you created, Event Coordination.

4. Click *Save*.

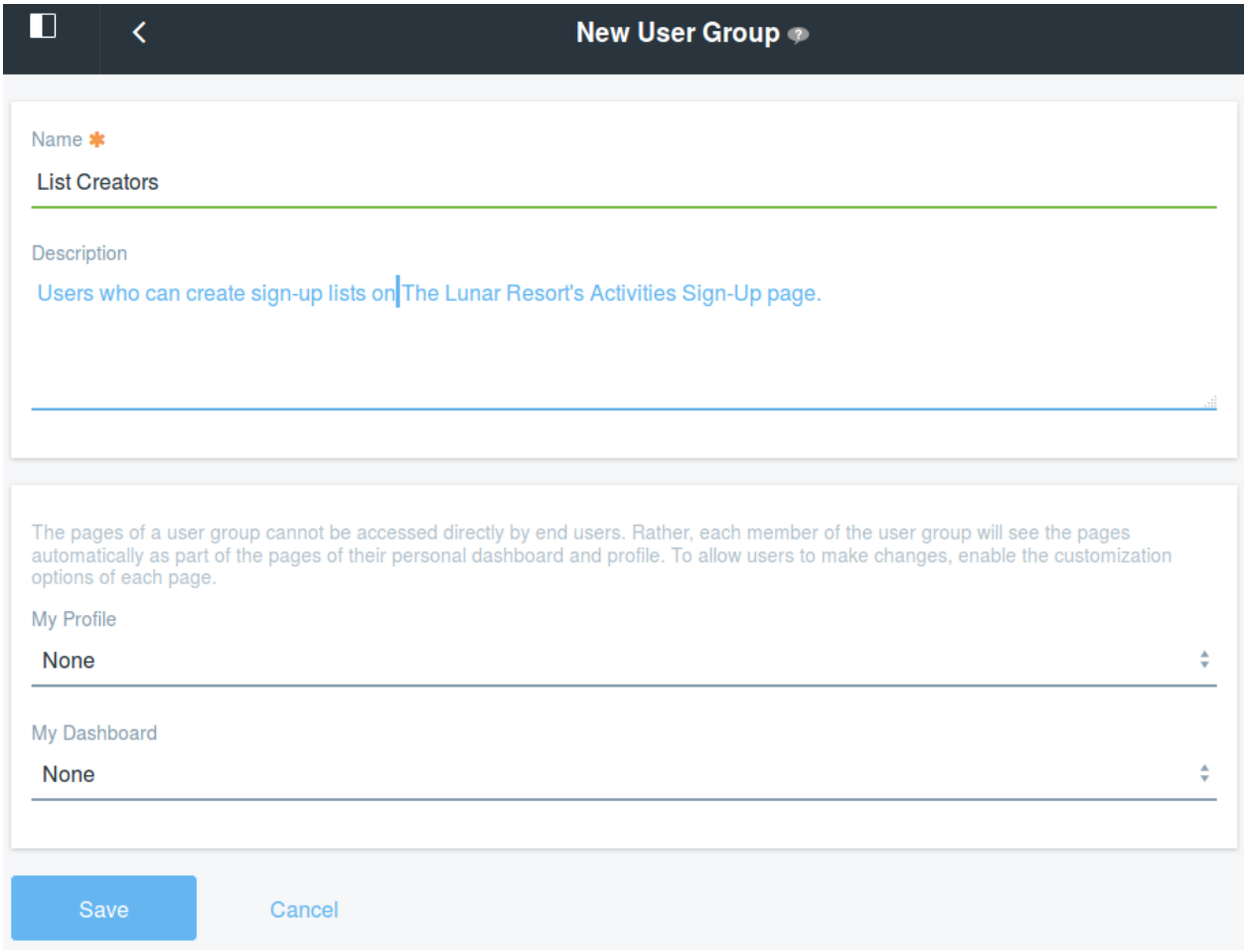


Figure 27.24: You can create a user group site from a site template while adding the User Group initially, or you can revisit and edit this form if you decide to add a site later.

Now, when one of the List Creators signs in to The Lunar Resort and clicks on their *My Dashboard* link, the Event Coordination page will be added to their private page set.

User group site pages function similarly to regular site template pages, with an important exception: user group site pages are not copied for each user. They're shown dynamically along with any custom pages that the user may have on his/her personal site. For this reason, users are not allowed to make any modifications to the pages that are inherited from the user group. If needed, the user group administrator can define certain areas of a page as customizable, like with regular sites. This allows users to add and configure applications in the specified area of the page.

You can create a user group's site manually, instead of basing it on a site template. To create a user group's site manually, use the *Actions* menu for the user group, as mentioned above, and choose *Manage Pages*. Add a new public or private page by selecting the Actions menu next to My Profile or My Dashboard under Site Administration for the Control Panel, then click the *Add Private Page* or *Add Public Page* link. Once the user group has at least one public or private page in place, you can go back to the *Actions* menu and click on the *Go to the Profile Pages* or *Go to the Dashboard Pages* link to open the user group's site in a new browser window. In the new window, you can add more pages and portlets and configure site settings.

In the example above, you based the List Creator user group's site on a template. Leave the *Enable propagation of changes from the site template* box checked to automatically update users' personal sites if the

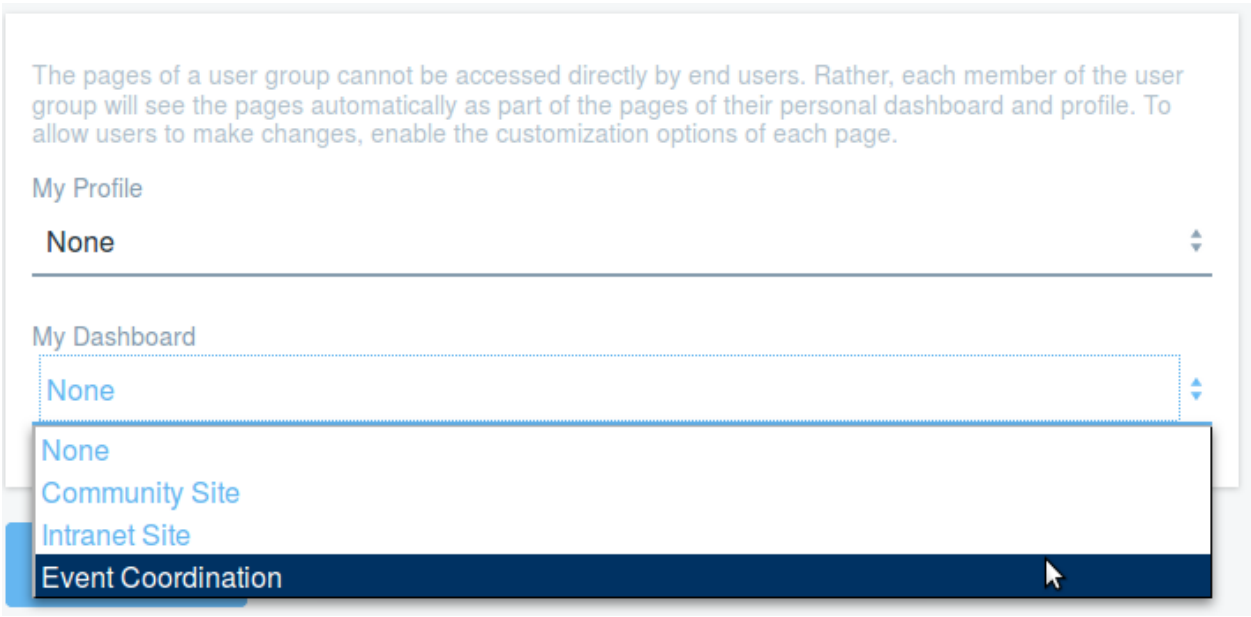


Figure 27.25: Selecting a site template under My Dashboard creates a private site for a user group.

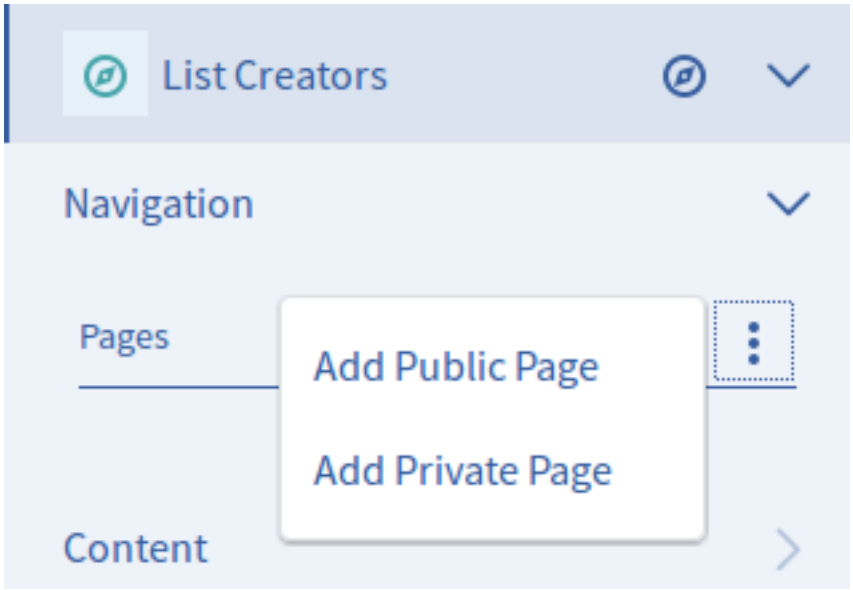


Figure 27.26: : From a user group's Actions menu, select *Manage Pages* to create a user group site manually.

associated site template changes. If you uncheck this box but recheck it later, the template pages are copied to the users' sites, overwriting any changes they may have made. You can allow users to make changes to the pages they receive from the user group by enabling the customization options on each page.

This flexibility lets you achieve almost any desired configuration for a user's personal site without having to modify it directly. When a user is assigned to a user group, they'll immediately have access to the user group's site pages from their personal site.

Legacy User Group Sites Behavior Since the inheritance of user group site pages is now dynamic, even if there are hundreds of thousands of users, even millions, there won't be an exponential impact in performance. Previous versions of Liferay required user group pages to be copied to each user's personal site. If you really liked the old days, or if you're upgrading from an older version of Liferay and need to keep the old behavior, enable it by adding the following line to your portal-ext.properties file:

```
user.groups.copy.layouts.to.user.personal.site=true
```

When this property is set to true, the template pages are copied to a user's personal site once, and then may be modified by the user. This means that If changes are made to the template pages later, they will only affect users that are added to the user group after the change is made. Users with administrative privileges over their personal sites can modify the pages and their content if the *Allow Site Administrators to Modify the Pages Associated with This Site Template* box has been checked for the template. When a user is removed from a user group, the associated pages are removed from the user's personal site. If a user is removed from a group and is subsequently added back, the group's template pages are copied to the user's site a second time. Note that if a user group's site is based on a site template and an administrator modifies the user group's site template after users have already been added to the group, those changes only take effect if the *Enable propagation of changes from the site template* box for the user group was checked.

Just like user groups simplify site membership, they can also be used to assign a role to all the users of a user group at once. Where you do it depends on the scope of the role.

Configuring User Group Permissions

The List Creators user group needs the ability to manage Dynamic Data List Display portlets on The Lunar Resort's Activities Sign-Up public page. This will require the creation of a new Site Role called List Creator, and assignment of any necessary permissions so the List Creators can do their job. You can refer to the article on Roles for a more detailed discussion. After creating the new role, all that's left to do is assign the role to the user group.

To create a new role, go to the Roles section of the Control Panel and click the *Site Roles* tab. Click *Add → Site Role*. Call it *List Creator*, give it a description, and save it. From the list of roles, open the Actions menu for your new role, and select *Define Permissions*. Select *Site Administration → Applications → Dynamic Data Lists Display*. Select just the *Configuration* and *View* permissions and click *Save.* Go back to the list of roles by clicking *Roles* from the Product Menu or by clicking the back arrow ( ) at the top left of the Define Permissions window.

Now you have a user group, and a site role that's ready to be used in The Lunar Resort site. The user group members are also members of The Lunar Resort site, so all that's left is to associate the user group with the role.

1. Go to The Lunar Resort's Site Membership section.

2. In the Site Memberships window, User Groups tab, open the configuration menu ( ) for the List Creators user group and select *Assign Site Roles*.

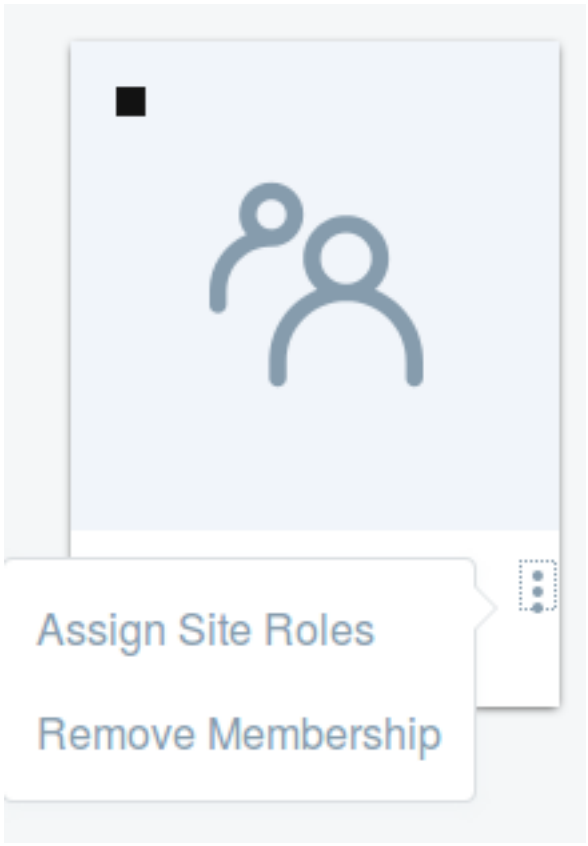


Figure 27.27: : From a user group's Actions menu, select *Manage Pages* to create a user group site manually.

3. Choose the *List Creator* role from the list and click *Done*.

Now members of the List Creators user group are site members of The Lunar Resort site, and anyone in the user group has the List Creator role in the site. If you want to continue configuring the List Creators user group, read the articles on Creating Simple Applications. In those articles you'll learn to create a data definition that the List Creators can use to build new Activities Sign-Up lists and templates with.

If you want to know more about User Groups, and how to edit them, keep reading here.

Editing User Groups

To edit the basic properties a user group, click the *Edit* link from the Actions menu of the user group. Alternatively, click on its name or description (both are hyperlinks), and then open the *Actions* menu in the top right and select *Edit*. Also in the *Actions* menu, you'll see the full list of actions that can be performed on a user group. When editing a user group, you can view its site, if it exists, by clicking the *Open Pages* link under Public Pages or Private Pages (read above for details on user group sites).

As with the other resources in the portal, you can click the Actions button (⋮) next to a user group to perform various operations on that group.

Edit allows you to modify the name, description, or choose the site template for a user group site.

Permissions lets you define which roles have permissions to view, edit, delete, assign members to the user group, etc.

User Group Pages Permissions lets you define which roles have various permissions related to managing the user group's site pages.

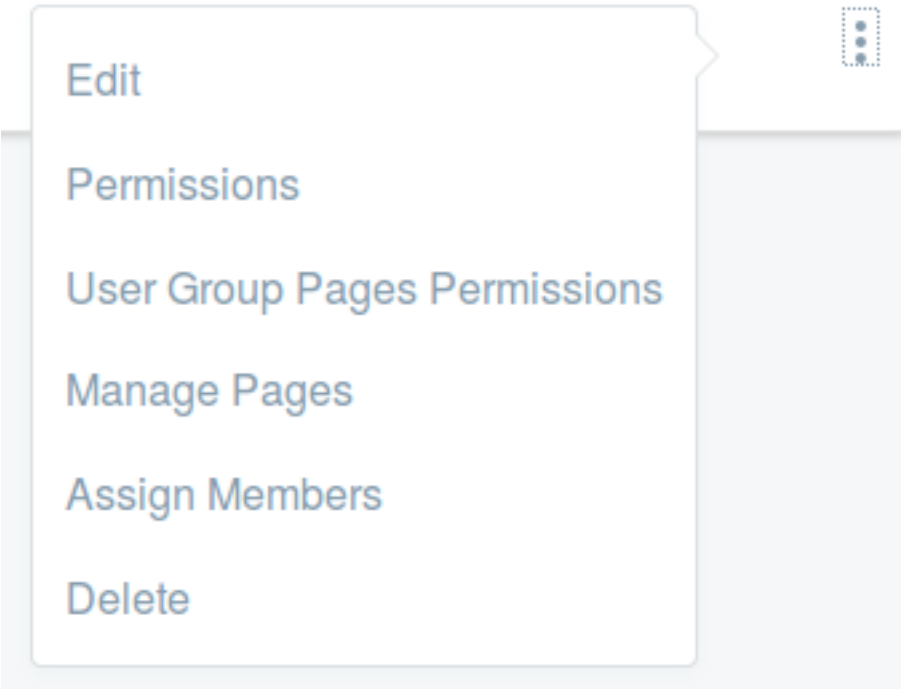


Figure 27.28: : The user group's Actions menu has various configuration options.

Manage Pages takes you to the page administration section of the user group's site. You can add pages to the user group site, import or export pages, organize the page hierarchy, modify the look and feel of the pages, add a logo or access other options from the Manage Site interface.

Assign Members lets you search for and select users in the portal to be assigned to this user group as well as view the users currently belonging to the user group.

Delete removes the user group.

Note: If there are users in the User Group, you won't be able to delete it. You'll first need to remove the users from the User Group.

If your user group has site pages, the options *Go to Profile Pages* and *Go to Dashboard Pages* also appear in your user group's Actions menu. Clicking one of these links opens the user group's site in a new browser window. Any changes you make to the site are saved automatically. You can safely close the browser window when you're done.

User groups are useful for managing site membership and roles, and for assigning a specific task to a specific group of users. Or, if you like tongue twisters, this article covered *the usual uses that use useful user groups*. Let us know if there's any functionality we missed, or if there's something that's not clear from this article, by leaving a comment in the *Do you have any suggestions?* field. Good jokes (okay, any jokes) can also be submitted.

27.4 Roles and Permissions

If a *role* were to win a Grammy or an Oscar or some other ego-feeding popularity contest, it better remember to thank all its *permissions* groupies during the acceptance speech, because they're the ones doing the real

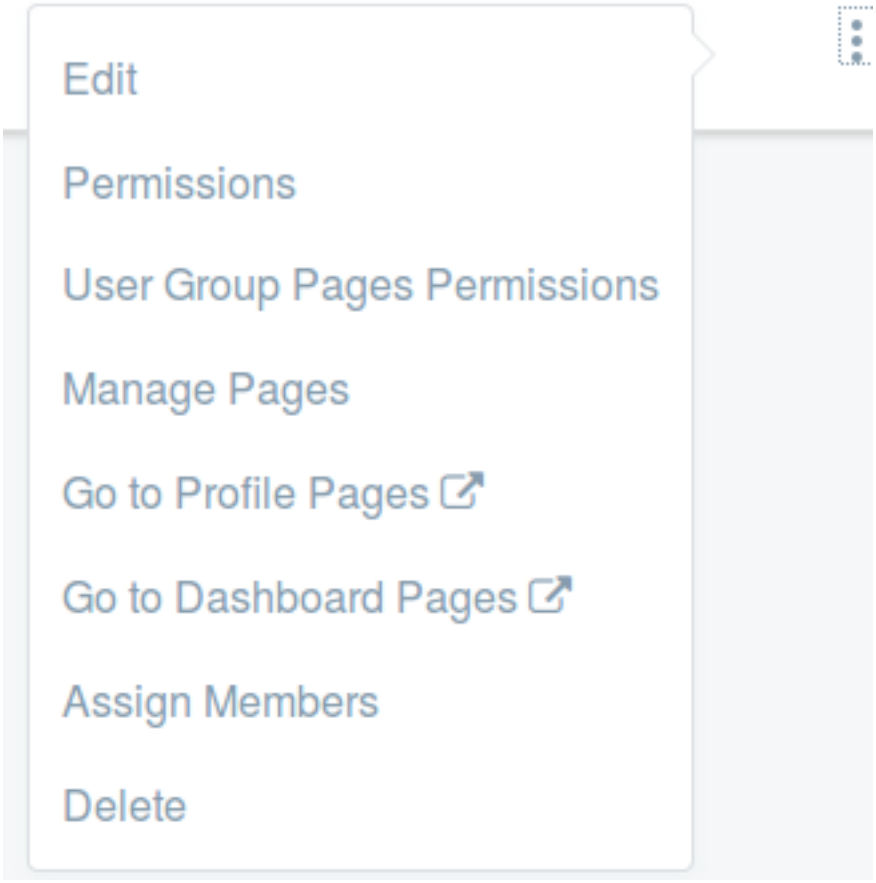


Figure 27.29: : If your user group has public and private site pages, you'll see *Go to Profile Pages* and *Go to Dashboard Pages* in the Actions menu.

work. The role is just the pretty face, so to speak.

Roles collect *permissions* that define a particular function within Liferay DXP, according to a particular scope. Roles collect permissions, and users are assigned to roles, either directly or through their association with a User Group, an Organization, or a Site.

Take a *Message Board Administrator* role, for example. A role with that name is likely to have permissions relevant to the specific Message Board portlets delegated to it. Users with this role inherit the permissions collected underneath the umbrella of the role.

NOTE: Roles are managed through the Control Panel, but not all permissions are. Asset-level permissions (for instance, permission to edit an individual blog post, or view a folder in the Documents and Media library) are managed from the individual asset. See Application Permissions for details.

Managing Roles in Liferay

Manage Liferay's roles in the Control Panel (*Control Panel → Users → Roles*). There you'll find an application for creating roles, granting them permissions, and assigning users to them. Roles can be scoped by portal, site, or organization.

To create a role, click the scope you want a role for and then click the *Add* ( ) button. Enter a name, title, and description for the role. The name field is required but the title and description are optional. If you enter

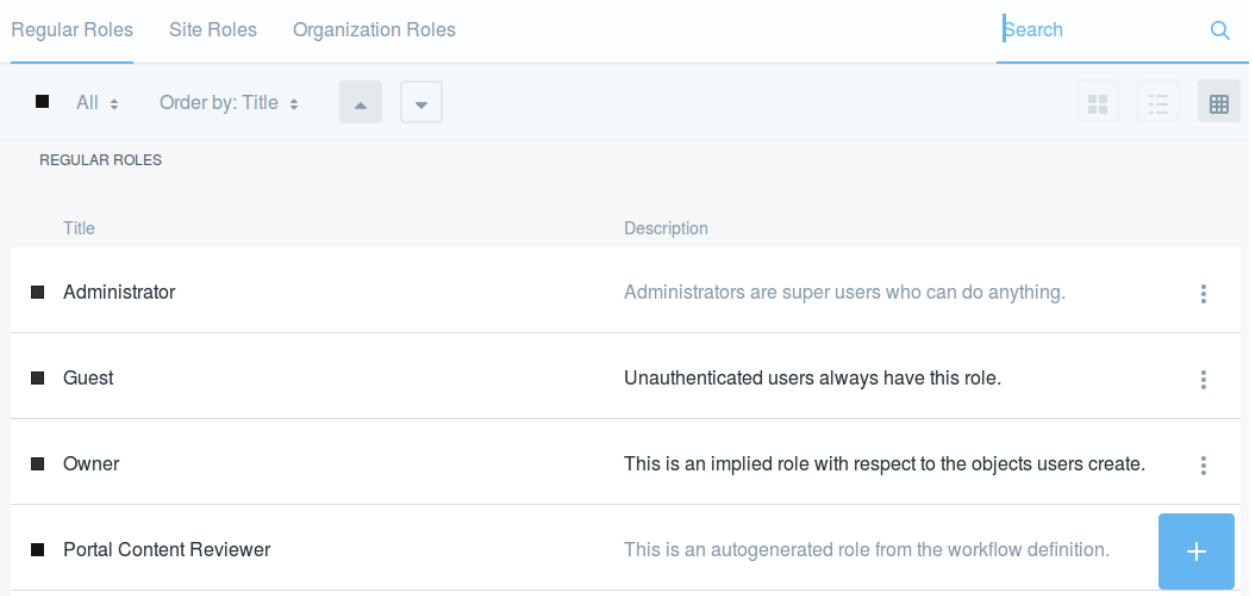


Figure 27.30: The Roles application lets you add and manage roles for the global (*Regular*), *Site*, or *Organization* scope.

a name and a title, the title is displayed in the list of roles on the Roles page of the Control Panel. If you do not enter a title, the name is displayed. When you finish, click *Save*.

Note: In addition to regular roles, site roles, and organization roles, there are also teams. Teams can be created by site administrators within a specific site. The permissions granted to a team are defined and applied only within the team's site. The permissions defined by regular, site, and organization roles, by contrast, are defined at the portal level, although they are applied to different scopes. The differences between the four types of roles can be described as follows:

- Regular role: Permissions are defined at the *portal* level and are applied at the *portal* level.
- Site role: Permissions are defined at the *portal* level and are applied to one *specific site*.
- Organization role: Permissions are defined at the *portal* level and are applied to one *specific organization*.
- Team: Permissions are defined within a *specific site* and are assigned within that *specific site*.

Read here For more information about teams.

After you save, your role is added to the list of roles. To see what functions you can perform on your new role, click the *Actions* button.

Edit: lets you change the name, title or description of the role.
Permissions: allows you to define which users, user groups or roles have permissions to edit the role.
Define Permissions: defines the permissions the role contains.
Delete: permanently removes a role from the portal.
Once you have a role you want to configure, the first step is often to define its permissions.

Defining Role Permissions

Roles collect permissions, so when a user is given a role, they receive all the permissions defined by the role.

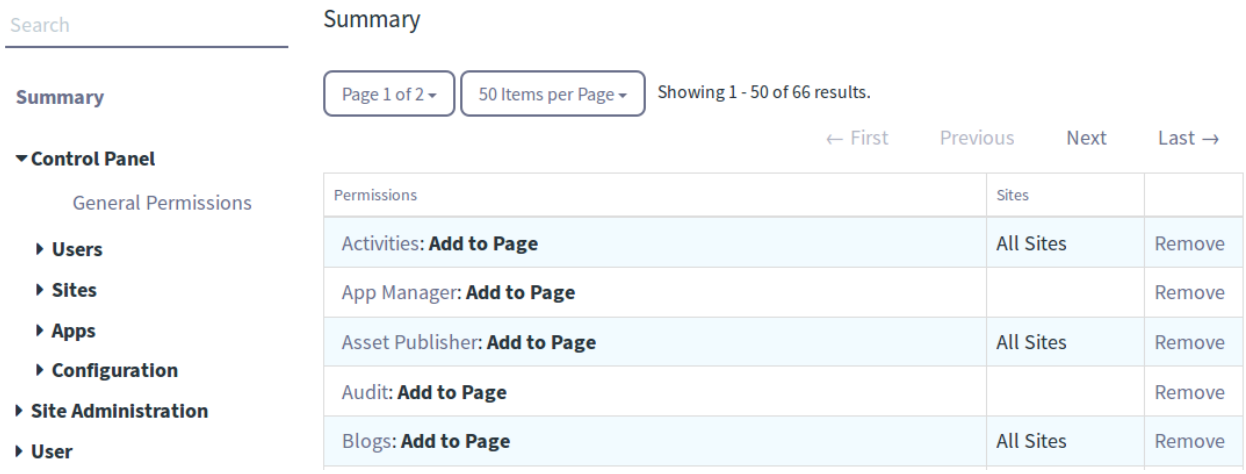


Figure 27.31: When defining permissions on a role, the Summary view provides a list of permissions that have already been defined for the role. The area on the left side of the screen lets you drill down through various categories of portal permissions.

To add permissions to a role, click on the *Actions* (⋮) button for a regular role and select *Define Permissions*. Find the permissions you want to add by navigating the categories of permissions on the left side of the screen and click on a specific category (such as *Site Administration* → *Navigation* → *Site Pages*). Select any permissions that you'd like to add the role, then click *Save*.

Note: The Roles application in the Control Panel is not the only place where permissions are configured. You can configure a role's permissions for a particular application instance from its *Options* (⋮) menu. However, permissions granted or removed in the Control Panel override those made at the more granular level.

There are three basic categories of permissions: *Control Panel*, *Site Administration*, and *User*. By default, any Liferay user can manage their user account via the permissions belonging to the User category. Site Administrators can access the site administration tools belonging to the Site Administration category. Portal Administrators can access the entire Control Panel. For custom roles, you can mix and match permissions from as many categories as you like.

The permissions in the Site Administration → Applications categories govern the content that can be created by core portlets such as the Wiki and Message Boards. If you pick one of the portlets from this list, you'll get options for defining permissions on its content. For example, if you pick Message Boards, you'll see permissions for creating categories and threads or deleting and moving topics.

Site application permissions affect the application as a whole. Using the Message Boards as an example, an application permission might define who can add the Message Boards portlet to a page.

The Control Panel permissions affect how the Control Panel appears to the user in the Control Panel. The Control Panel appears differently to different users, depending on their permissions. Some Control Panel portlets have a Configuration button and you can define who gets to see that. You can also fine-tune who gets to see various applications in the Control Panel.

If you want to change the scope of a permission, click the *Change* link next to the gear icon next to the permission and then choose a new scope. After you click *Save*, you'll see a list of all permissions currently

granted to the role. From the Summary view, you can add more permissions or go back to the Role Application default view by clicking on the *Back* ( ) icon.

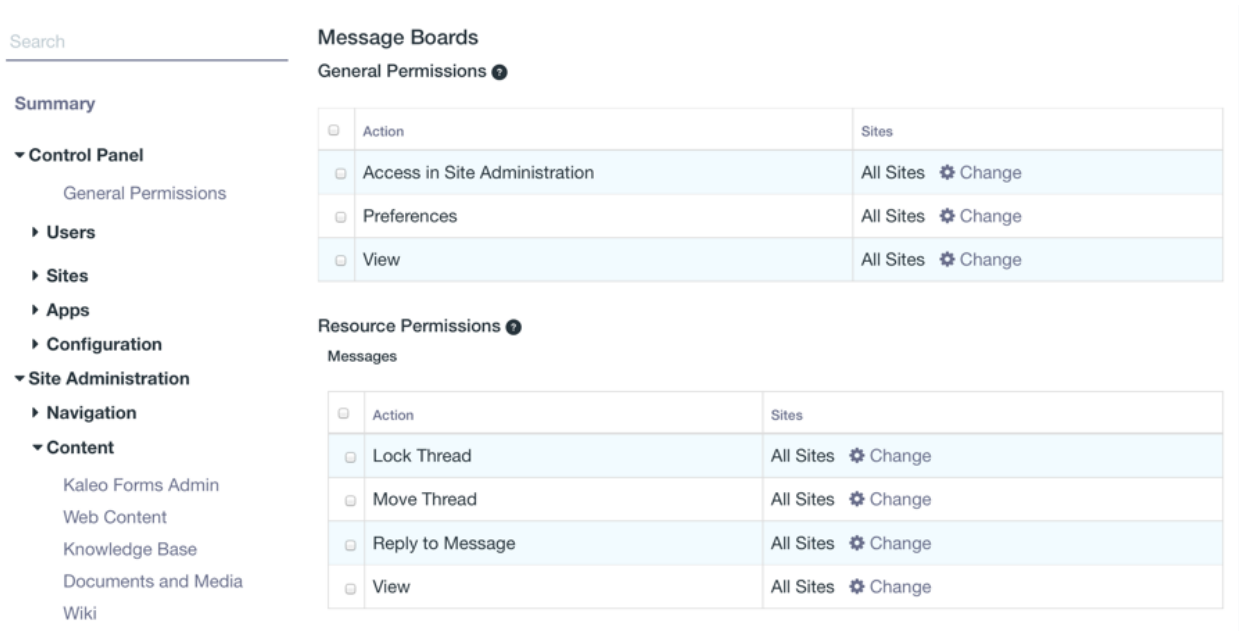


Figure 27.32: You can fine-tune which actions are defined for a role within a specific application like the Message Boards.

Sometimes you might find that a certain permission grants more or less access than what you expected–always test your permissions configurations!

Suppose you need to create a role called *User Group Manager*. You'd like to define the permissions for the User Group Manager role so that users assigned to this role can add users to or remove users from any user group. To do this, you can take the following steps:

1. Go to the Control Panel and then click on *Users* → *Roles*.
2. On the Regular Roles screen, click *Add* ( ).
3. After naming your role and entering a title, click *Save*.
4. Click on *Actions* ( ) → *Define Permissions* and drill down in the menu on the left to *Control Panel* → *Users* → *User Groups*.
    - Under the *General Permissions* heading, flag *Access in Control Panel* and *View*. This lets user group managers access the User Groups Control Panel portlet and view existing user groups.
    - Since you'd like user group managers to be able to view user groups and assign members to them, you'd also check the *Assign Members* and *View* permissions under the *Resource Permissions* → *User Group* heading.
5. Click *Save*.

Once you create the role, assign it to its intended users. To assign roles to Users, Sites, Organizations, and User Groups, click on the role, then click on the Add button ( ). Choose the users and/or groups you want to assigned to the role. If assigning a group, note that all users assigned to that group will inherit the role as well.

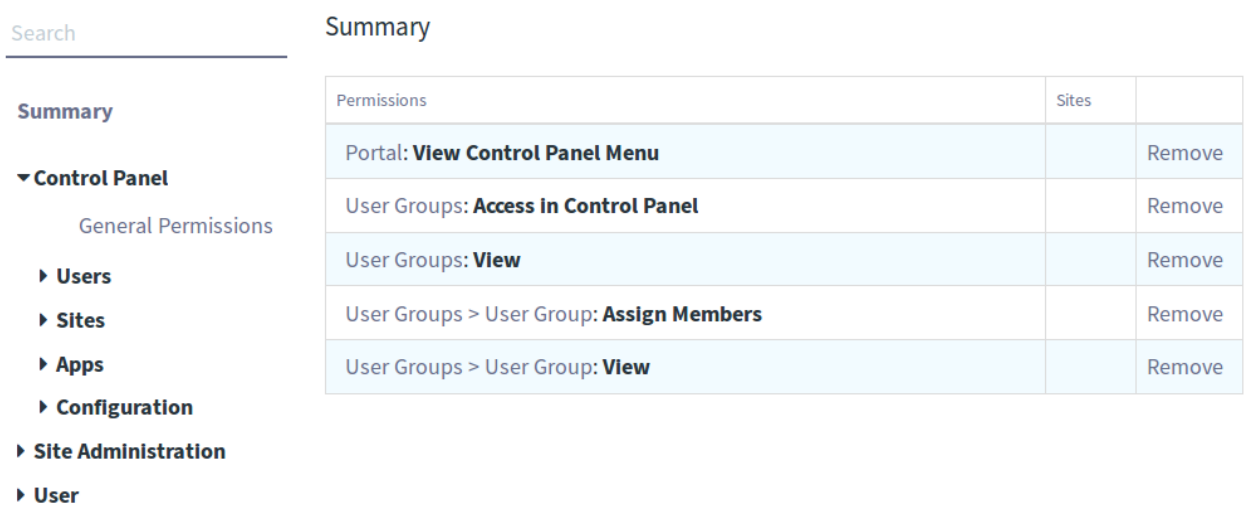


Figure 27.33: Make sure to test the permissions you grant to custom roles.

You might expect that the role has all the permissions necessary for adding users to user groups. After all, user group managers can view user groups, assign members, and access User Groups in the Control Panel. However, we're forgetting an important permission: the User Group Manager role can't view users! This means that if they click *Assign Members* for a user group and click on the *Available* tab, they'll see an empty list.

If you create a role with permission to access something in the Control Panel, keep in mind that the *View Control Panel Menu* permission will be automatically granted. Consider why this is necessary with an example.

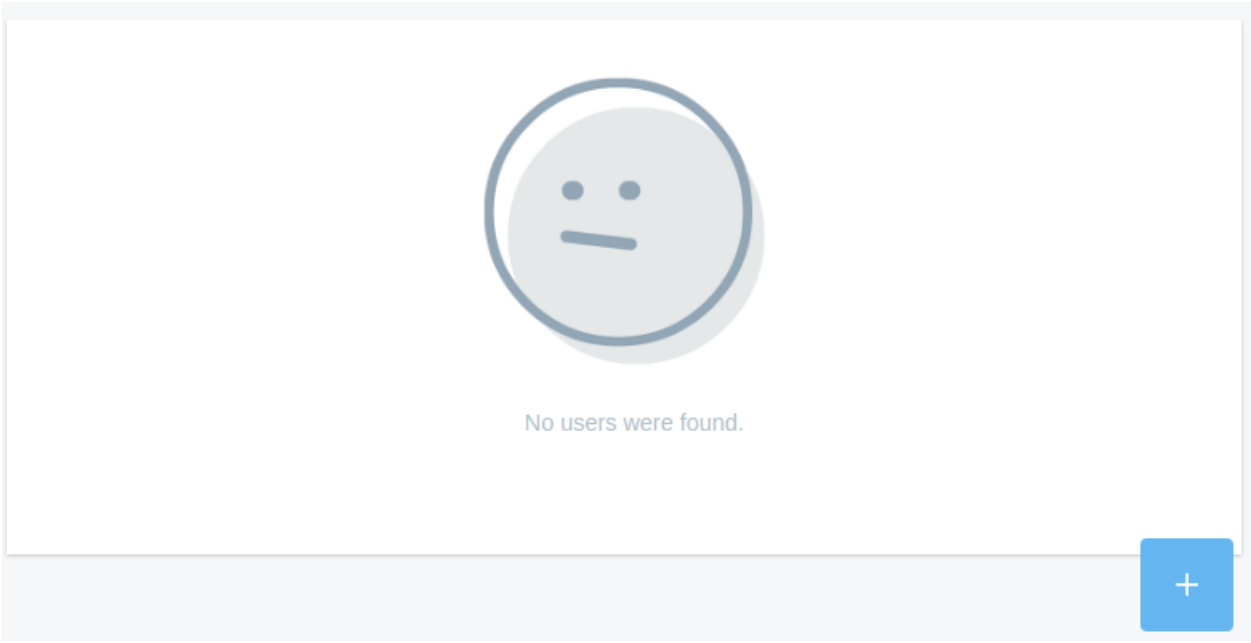


Figure 27.34: Users assigned to the User Group Manager role can't find any users to add!

To fix this, define the missing permission on the role by drilling down to the *Control Panel → Users → Users and Organizations* category and flag the *View* permission under the *Resource Permissions → User* heading. Once you've saved your permissions configuration, users who've been assigned to the User Group Manager role will be able to browse the portal's entire list of users when assigning users to a user group.

Roles are very powerful and allow portal administrators to define various permissions in whatever combinations they like. This gives you as much flexibility as possible to build the site you have designed.

Permission for Delegating Social Activities Configuration

There's a permission that allows site administrators to delegate responsibility for configuring social activities to other users. To add this permission to a role, click *Actions* next to the desired role and select *Define Permissions*. Find the *Site Administration → Configuration → Social Activity* permissions category. Flag all of the permissions and then click *Save*:

- Access in Site Administration
- Configuration
- Permissions
- Preferences
- View

Once these permissions are assigned, assignees can manage the site's Social Activities.

Deleting Asset Containers

A Web Content Folder contains Web Content articles. The Web Content Folder is an asset container, and the Web Content Article is an asset. It's possible to give a role permission to delete an asset container without giving the role permission to delete individual assets. In that case, beware: if a role assignee deletes an asset container with individual assets in it, the individual assets themselves *will* be deleted as well.

Besides Web Content Folders, examples of asset containers include Bookmarks Folders, Message Boards Categories, Wiki Nodes, and Documents and Media Folders.

You might not need to create a role for a certain functionality. Liferay provides many pre-configured roles for your convenience.

Default Liferay Roles

In the Roles Application, you'll see a list of all the roles in Liferay, by scope. These are some of the pre-configured roles:

- Guest: The Guest role is assigned to unauthenticated users and grants the lowest-level permissions within the portal.
- User: The User role is assigned to authenticated users and grants basic basic permissions within the portal (mostly *Add to Page* permissions for applications).
- Power User: The Power User role grants more permissions than the User role. It's designed to be an extension point for distinguishing regular users from more privileged users. For example, you can set up your portal so that only Power Users have personal sites.
- Site Member: The Site Member role grants basic privileges within a site, such as the ability to visit the site's private pages.

- Site Administrator: The Site Administrator role grants the ability to manage *almost* all aspects of a site including site content, site memberships, and site settings. Site Administrators cannot delete the membership of or remove roles from other Site Administrators or Site Owners. They also *cannot* assign other users as Site Administrators or Site Owners.
- Site Owner: The Site Owner role is the same as the Site Administrator role except that it grants the ability to manage *all* aspects of a site, including the ability to delete the membership of or remove roles from Site Administrators or other Site Owners. They *can* assign other users as Site Administrators or Site Owners.
- Organization User: The Organization User role grants basic privileges within an organization. If the organization has an attached site, the Organization User role implicitly grants the Site member role within the attached site.
- Organization Administrator: The Organization Administrator role grants the ability to manage *almost* all aspects of an organization including the organization's users and the organization's site (if it exists). Organization Administrators cannot delete the membership of or remove roles from other Organization Administrators or Organization Owners. They also *cannot* assign other users as Organization Administrators or Organization Owners.
- Organization Owner: The Organization Owner role is the same as the Organization Administrator role except that it grants the ability to manage *all* aspects of an organization, including the ability to delete the membership of or remove roles from Organization Administrators or other Organization Owners. They *can* assign other users as Organization Administrators or Organization Owners.
- Administrator: The administrator role grants the ability to manage the entire portal, including global portal settings and individual sites, organizations, and users.

Tip: It's easy to overlook the differences between owner type roles and administrator type roles for sites and organizations. Site and organization administrators *cannot* remove the administrator or owner role from any other administrator or owner, and they *cannot* appoint other users as site or organization administrators or owners.

In contrast, site and organization owners *can* do those things.

Roles, and the permissions granted with their assignment, are foundational components in Liferay. Understanding their uses and configuration should enhance your ability to configure Liferay DXP to suit your organizational needs.

Chapter 28

GDPR Tools

Internet users are increasingly and justifiably concerned about how their personal data is processed by the systems they use. The enforcement of GDPR is a crystallization of these concerns into legislative action. Companies processing the personal data of EU residents must adopt appropriate measures to protect User data. Of course, legal requirements like those in GDPR only explain one reason for companies to develop policies for ensuring their users' right to privacy. The market demands site owners show higher levels of responsiveness to User inquiries into how their data is stored and processed.

Liferay is aware of the need for functionality to address User data management, and as of Fix Pack 60 has added two important features toward this end:

Erase and/or anonymize data associated with a User Administrative Users go through a step by step process, choosing to erase certain pieces of data and anonymize others.

Export a User's personal data Export ZIP files containing the data associated with a User.

These features are tools that get you closer to meeting two of GDPR's technically challenging requirements, the *right to data portability* and the *right to be forgotten*.

Note: It is Liferay's sincerest hope that through the User Management functionality of Liferay DXP, companies processing the personal data of their website's users can satisfy the requirements of GDPR. However, the tools discussed here and anywhere else in the documentation, including those directly aimed at addressing GDPR requirements (as in this article) do not guarantee compliance with the legal requirements of GDPR. Each company or individual whose website processes user personal data and is under the jurisdiction of GDPR must carefully determine the precise steps necessary to ensure they are fully compliant with GDPR.

28.1 Anonymizing Data

Deleting User data is the safest way to honor *right to be forgotten* requests. When User data must be preserved, automatic anonymization of the data is in order. Users being anonymized must have their identifiers (for example, User ID and User Name) removed from content they've interacted with. However, portal content usually requires this information for its applications to work properly. Therefore, the User's identifiers must be replaced by something, or someone. Meet the new User, *Anonymous Anonymous*, identity swapper *extraordinaire*. This deactivated User is dedicated to be the User whose identifiers are assigned to anonymized

content. This identity swap is an important step in the anonymization process, but additional manual intervention may be necessary to truly achieve anonymization.

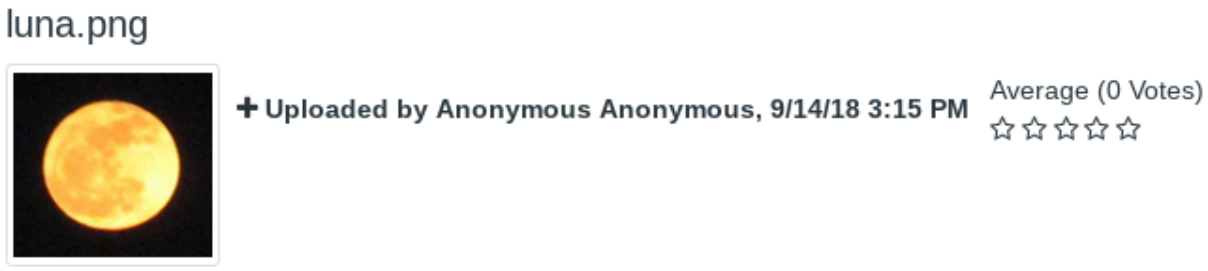


Figure 28.1: Anonymized content is presented with the User Anonymous Anonymous's identifying information.

Here at Liferay, we've grown fond of *Anonymous Anonymous*. If you'd rather start from scratch or assign an existing User to be the Anonymous User, get rid of *Anonymous Anonymous* and configure your own Anonymous User.

The anonymous user is programmatically created for each instance the first time an Administrator clicks *Delete Personal Data* from a User's Actions menu (⋮). If you haven't yet done that, no Anonymous User exists.

The easiest way to set up a new User as the Anonymous User is to edit an existing Anonymous User configuration, passing in the new Anonymous User's User ID.

To edit an existing configuration:

1. Go to *Control Panel* → *Configuration* → *System Settings*. Click on the *Users* tab and select *Anonymous User*.

2. Edit the existing configuration, providing a different User ID.

   Get the User ID from *Control Panel* → *Users* → *Users and Organizations*. Click on the User and find the User ID in the User Information screen of the Edit User application.

3. Click *Update*.

To create a new Anonymous User:

1. Create a User use for data anonymization. Alternatively, you can use an existing User.

2. If there's already an Anonymous User configured for the instance, there are two ways to remove it:

   Delete the User entirely. Deleting the User simultaneously deletes its configuration as the Anonymous User. Go to *Control Panel* → *Users* → *Users and Organizations*. If it's an active User, first deactivate, then delete the User. The default Anonymous Anonymous User is deactivated by default. Simply delete the User in this case. Click the Actions button (⋮) and select *Delete*.

   If you don't want to delete the User, just delete the User's configuration as the Anonymous User. Go to *Control Panel* → *Configuration* → *System Settings* → *Users* → *Anonymous Users*.

3. Add a new Anonymous User configuration. Click the *Add* button.

4. Fill out the two fields, Company ID and User ID.

   Get the Company ID from *Control Panel → Configuration → Virtual Instances*. The Instance ID and Company ID are the same.

   Get the User ID from *Control Panel → Users → Users and Organizations*. Click on the User and find the User ID in the Information screen of the Edit User application.

There can only be one Anonymous User configured for each instance.

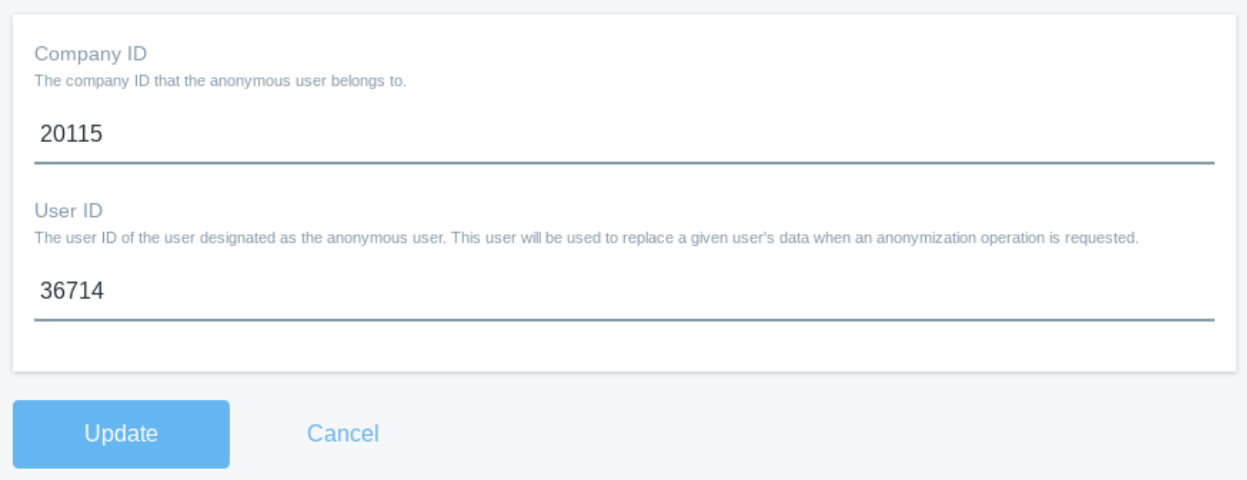


Figure 28.2: Assign your own Anonymous User from Control Panel → Configuration → System Settings → Users → Anonymous User.

28.2 Manual Anonymization

Anonymizing just the User's identification fields is often not enough. If a User named Ziltoid Omniscient complains about The Lunar Resort's coffee in a Message Boards Message and in it signs the post with *Supreme Leader of Ziltoidea 9*, anonymizing this post would remove the User's name (Ziltoid Omnisicent) and replace it with Anonymous Anonymous, but searching the internet for *Ziltoidea 9* quickly reveals that the post was written by Ziltoid the Omniscient. There can be user-entered personal data within the content of an application. You must manually edit such content to remove identifying details.

28.3 Sanitizing User Data

One of the technically challenging requirements of the General Data Protection Regulation (GDPR) is *the right to be forgotten*. The purpose of this article is not to go into the details of this requirement, but to show you how the personal data erasure functionality can assist you in satisfying this requirement.

A simple way to think of what it means to be *forgotten* by software is to consider a scenario where a new portal administrator is hired immediately after a User's right to be forgotten request has been honored. The new portal administrator has access to all of the Site's content and administration capabilities. Despite this, the administrator must not be able to glean information that could lead her to knowing the identity of the User whose personal data was erased.

Conceptually, forgetting a User means two things, at a minimum:

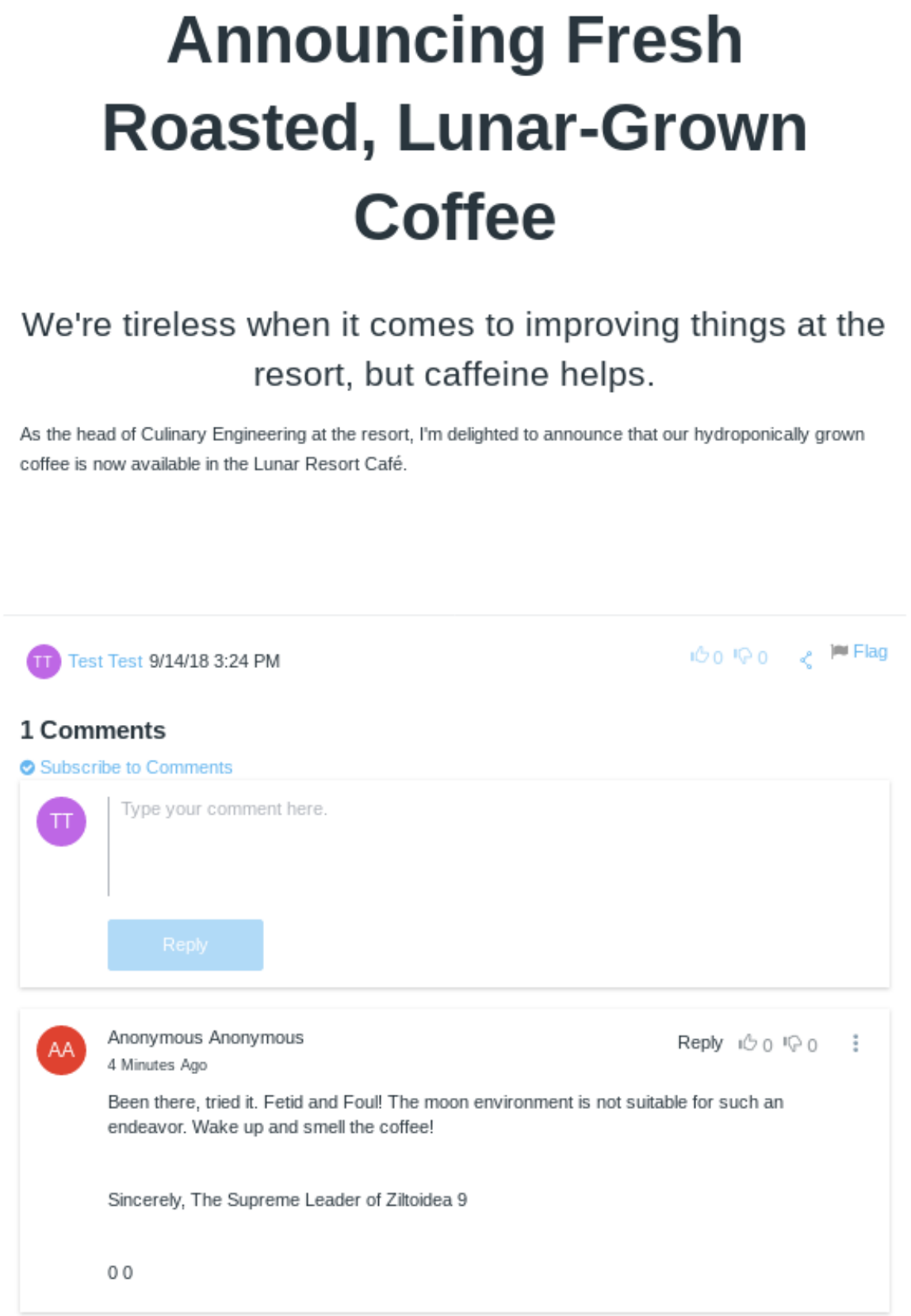


Figure 28.3: Even though this Message Boards Message (a comment on a blog post in this case) is anonymized, it should be edited to remove User Associated Data from the content of the message.

- Erasing the User's identifying information from the system. In Liferay DXP, this entails removing the User from database tables and search indexes.
- Erasing or anonymizing content the User has interacted with so it cannot be tracked to a real person.

Users can already be deactivated and then deleted, so why add new functionality? Deleting removes the User from the table of Users in the database. The User's information is preserved in other locations, however. In a standard User deletion scenario, all of a User's personally created content is still assigned to the User and her identifiers (User ID and User Name) still appear in the UI next to content associated with her. This unintentional preservation of user-identifying data is inadequate for satisfying some of the GDPR requirements and is the primary reason why the data erasure functionality was added in 7.0.

Note: Personal data erasure can help companies in their attempts to satisfy the requirements of GDPR. Using the data erasure tool described here provides no guarantee of compliance with the legal requirements of GDPR. Each company or individual whose website processes user personal data and is under the jurisdiction of GDPR must carefully determine the precise steps necessary to ensure they are fully compliant with GDPR.

To begin sanitizing a user's data,

1. Go to *Control Panel → Users → Users and Organizations*.
2. Click the *Actions* button for a User (⋮) and select *Delete Personal Data*.

   The User's Personal Data Erasure screen appears.
3. Complete the five-step process sequentially to erase the personal data associated with the User.

Step 1: Deactivation

Deactivating the User first ensures she doesn't create more content as you're sanitizing her from the system. Click *Deactivate User*.

Step 2: Delete the User's Personal Site

The User's public (profile) and private (dashboard) pages are deleted when the User is deleted. Separating this step out allows the administrator to make sure no information important to the enterprise is lost before the personal Site deletion is completed. Review the User's personal Site (click the provided links to navigate directly there) and preserve any necessary data. Then click *Delete Personal Site*.

Step 3: Review Application Data

There's no automated process for anonymizing application data (Blogs Entries, Wiki Pages, etc.). The administrator must review the User Associated Data (UAD: application content created by the User) piece by piece to determine that no data important to the enterprise is lost by deleting it and that no UAD can be gleaned from the content if anonymized.

After reviewing each piece of the data, the administrator either anonymizes it or deletes it to complete this step.

To enter the review process, click *Review*.

The Application Data Review screen displays a summary including how many content items in each application are associated with the User.

To manage (anonymize or delete) all the items for an application at once:

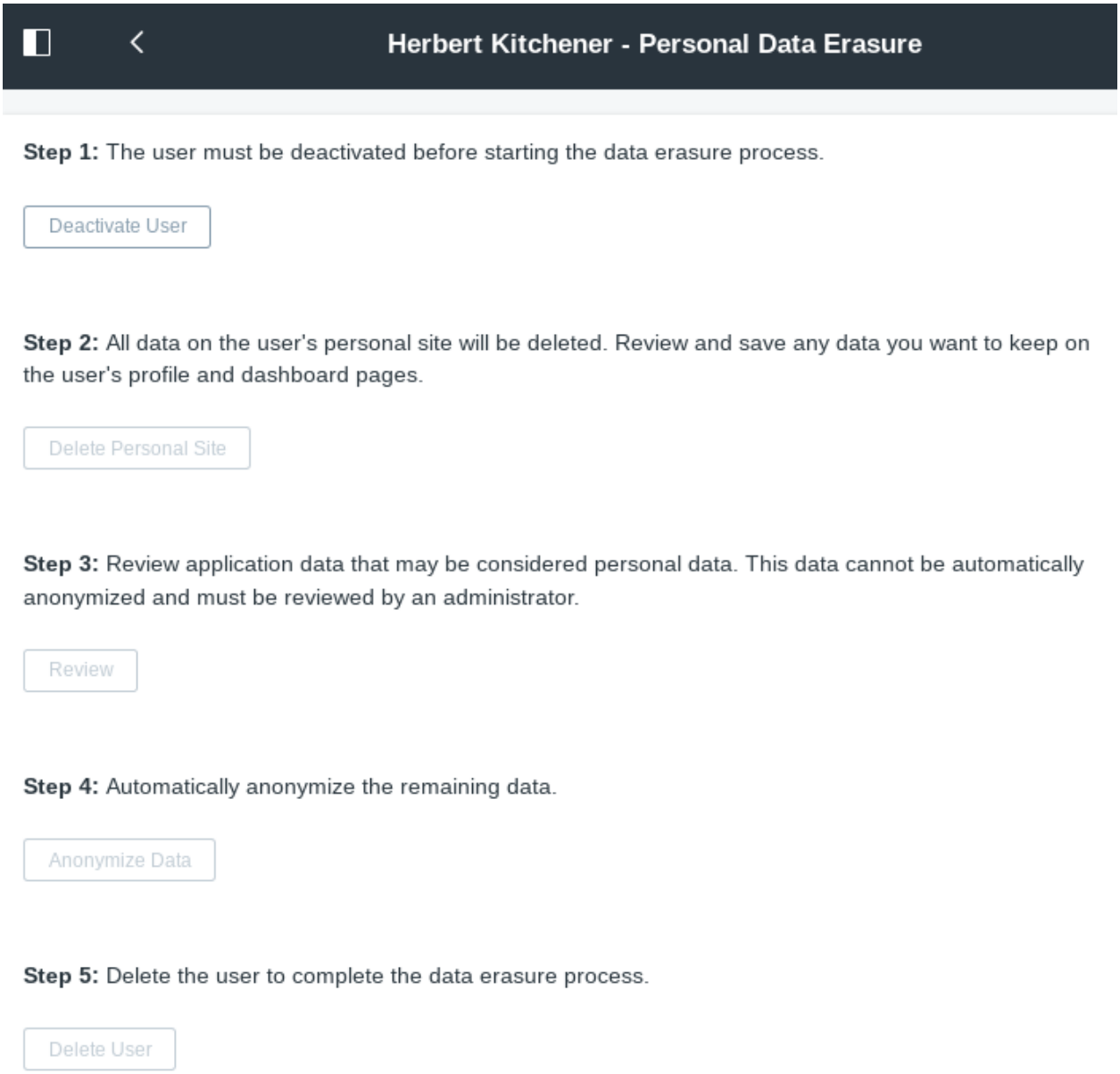


Figure 28.4: Sequentially complete five steps to complete the data erasure process.

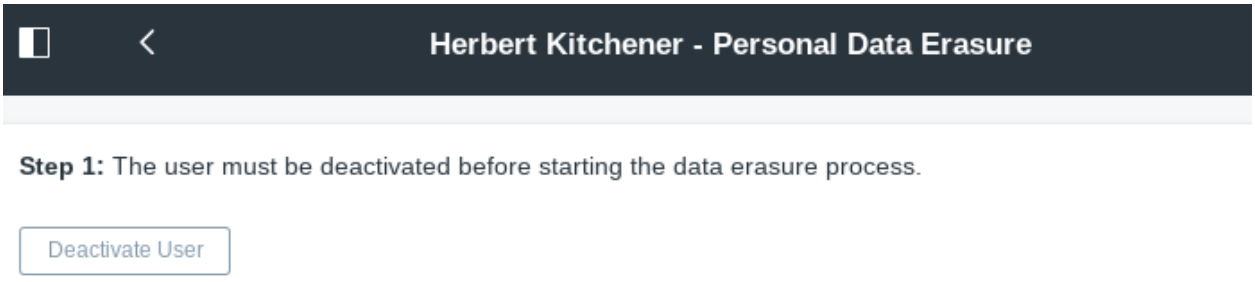


Figure 28.5: Deactivating the User kicks off the data erasure process.

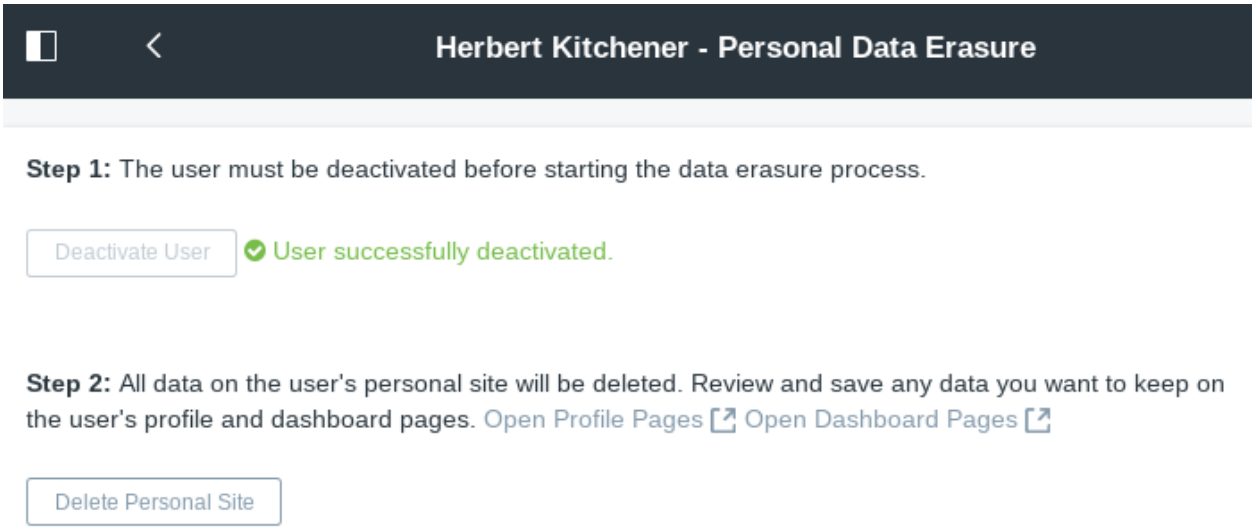


Figure 28.6: The second step in personal data erasure is deleting the User personal Site.

1. Click the Actions button (⋮) for the application.

2. If you're sure all items for an application can be safely deleted, choose *Delete*.

3. If you're sure simple anonymization is good enough for all of an application's items, choose *Anonymize*.

4. To view the items for an application, choose *View* or click on the application in the table.

Clicking an item takes you to the view/edit screen where you can see the application's items and take action.

Click the Actions button (⋮) for an item and select *Edit*, *Anonymize*, or *Delete*, as appropriate.

Once you've worked your way through the items and taken action, the view in the Application Data Review screen is updated to reflect that there are no more items needed to review. Click the *Complete Step* button once finished.

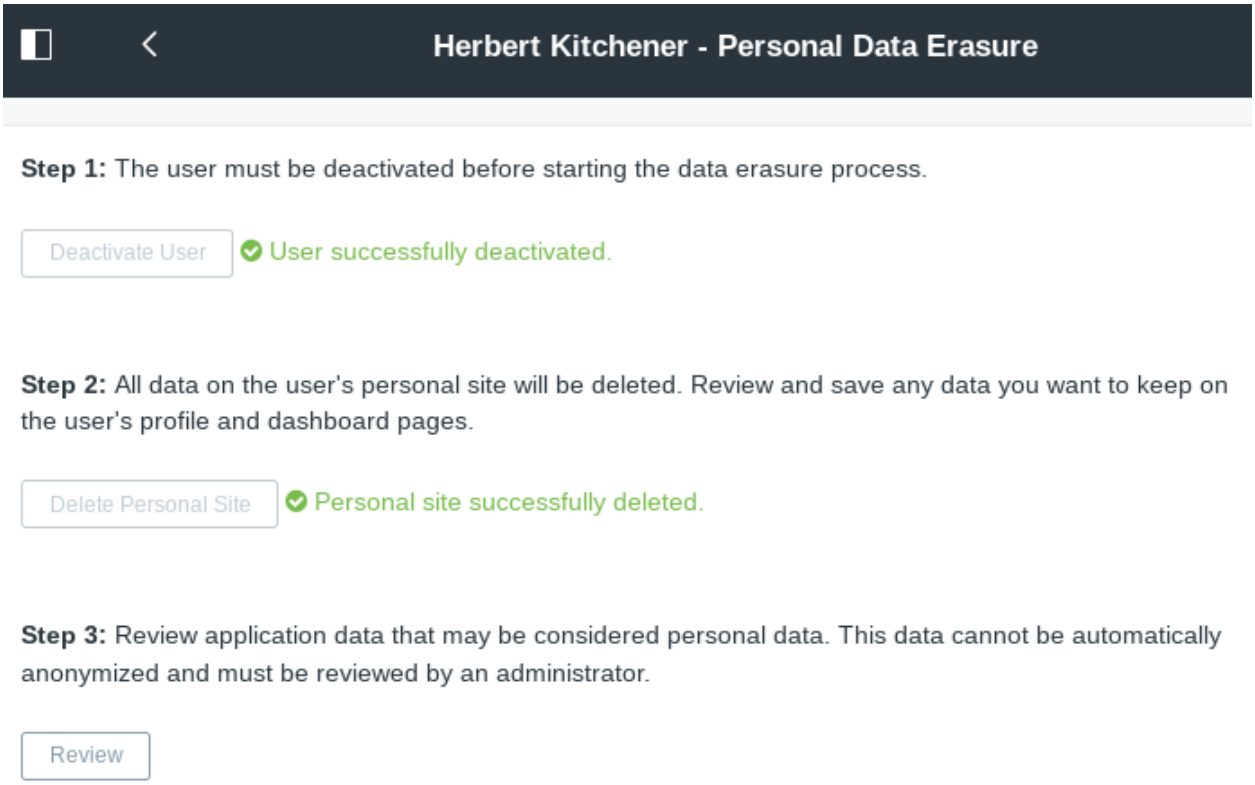


Figure 28.7: In the third data erasure step, review, delete, and/or anonymize the User's application data.

Step 4: Anonymize Remaining Data

In step 4, you must click *Anonymize Data*. This completes the anonymization process for remaining database references to the User's Name and ID. Some information is anonymized, but others, such as Notifications, are deleted as they don't make sense once the User is deleted.

Step 5: Delete User

Once all data is reviewed, deleted, edited, and/or anonymized as appropriate, delete the User. This step is simple: Click *Delete User*.

Now the User's data is anonymized or deleted, and the User is also deleted.

28.4 Exporting User Data

User Management practices must account for the EU's General Data Protection Regulation. One of its tenets is that Users have a right to *data portability*.

Data portability means that a User has the right to receive their personal data in a machine-readable format.

Note: Personal data export can help companies in their attempts to satisfy the requirements of GDPR. Using the export tool described here provides no guarantee of compliance with any GDPR requirement.

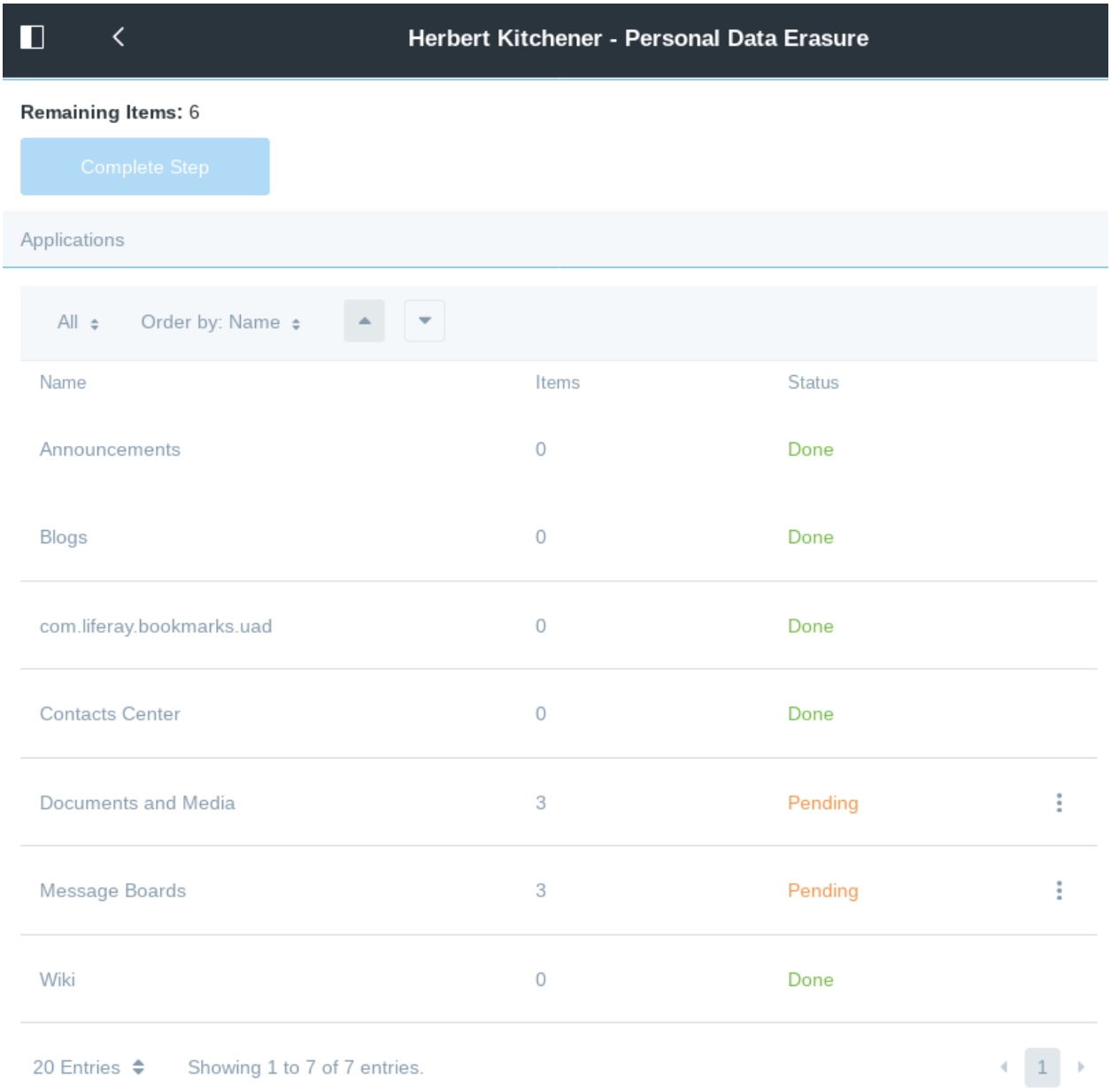


Figure 28.8: The Application Data Review screen makes the third, and most complicated data erasure step, more manageable.

Figure 28.9: Drill down into a User's application data to determine if manual editing is necessary.

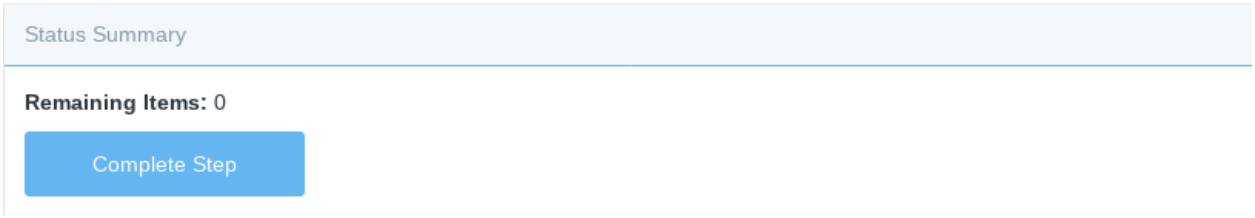


Figure 28.10: Click *Complete Step* once all of the User's application data is reviewed and dealt with.

Each company or individual whose website processes user personal data and is under the jurisdiction of GDPR must carefully determine the precise steps necessary to ensure they are fully compliant with GDPR.

The Control Panel's User Management system now supports the export of a User's personal data to a ZIP file for download. The data format for the files containing the data is XML.

Exporting and Downloading

To export user data,

1. Go to *Control Panel → Users → Users and Organizations*.
2. Find the User and click the *Actions* button ( ), then click *Export Personal Data*.

   This opens the User's *Export Personal Data* screen.
3. If there are no existing export processes shown, there's only one thing to do: click the *Add* button ( ). The tool for exporting the User's data appears.
4. Most of the time you want to export all the available data. Click the checkbox at the top of the screen, and all applications containing User data are selected in the UI.

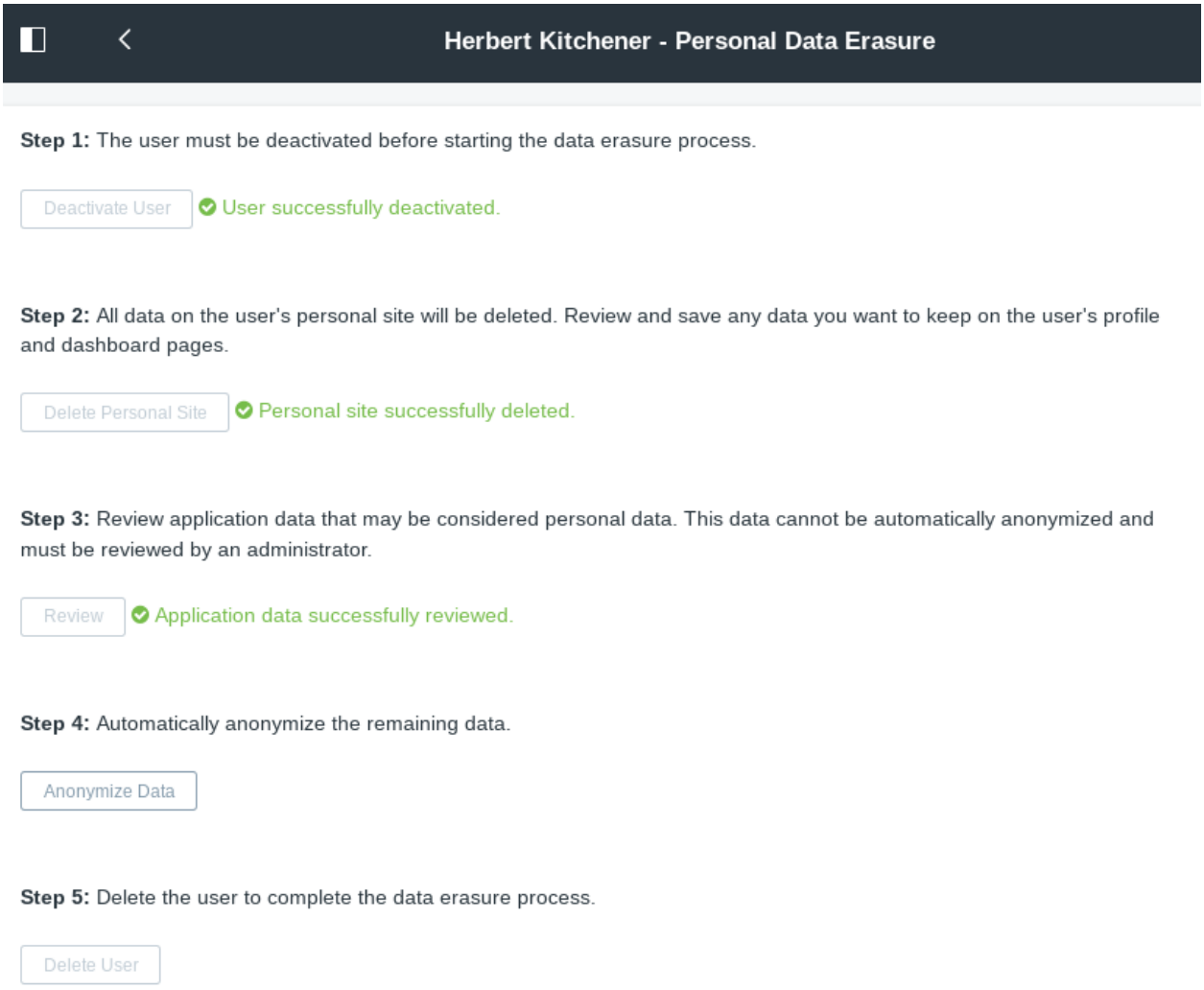


Figure 28.11: Anonymize any remaining use of the User's identifier's remaining in the database.

5. Click *Export*. You're taken back to the User's Export Personal Data screen, but now there's an export process in the list.

6. Download the data. Click the Actions button (⋮) for the process and select *Download*.

Examining Exported Data

So what does the exported data look like?

```
<?xml version="1.0"?>

<model>
    <model-name>com.liferay.message.boards.kernel.model.MBMessage</model-name>
    <column>
        <column-name>messageId</column-name>
        <column-value><![CDATA[37046]]></column-value>
    </column>
```

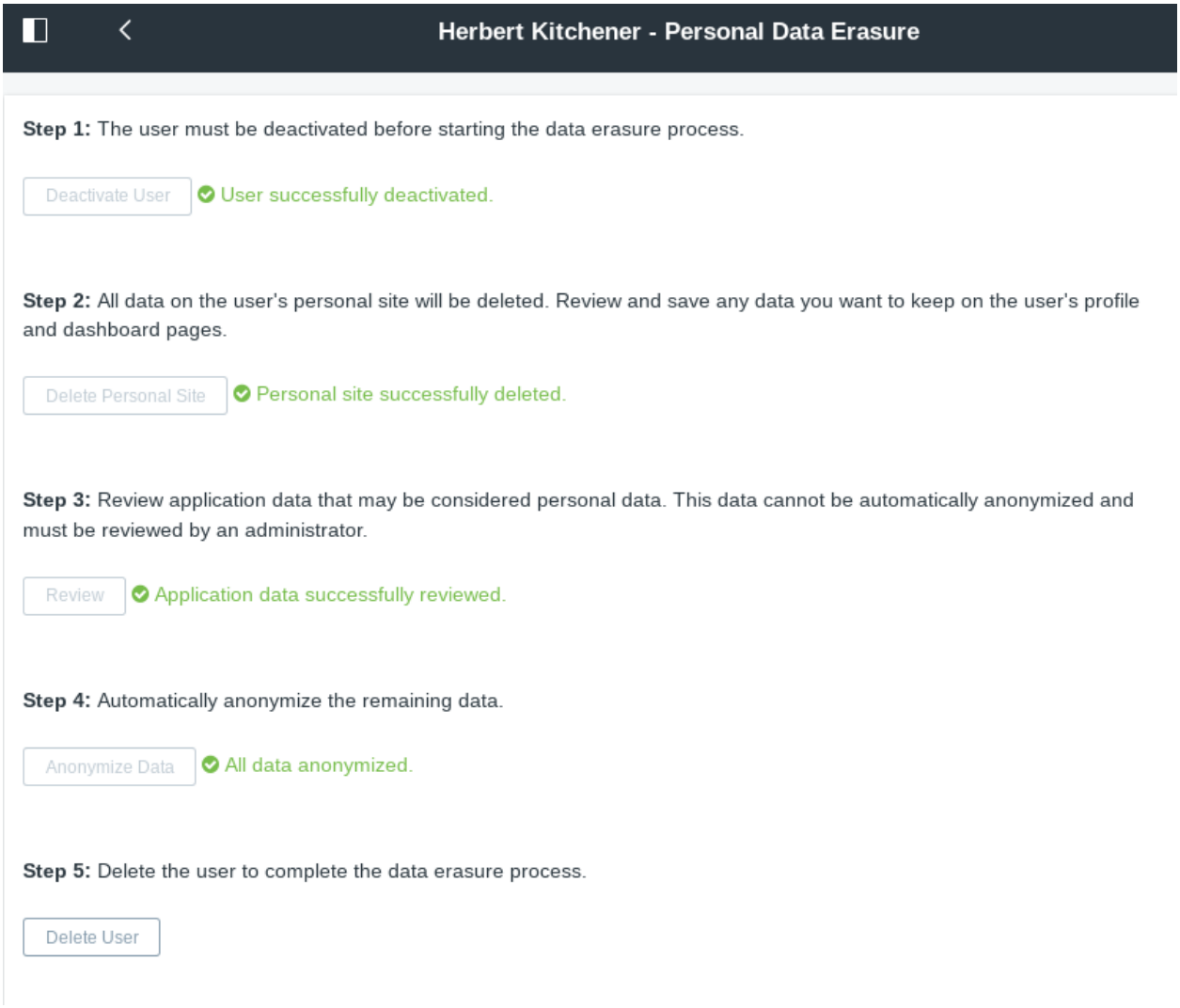


Figure 28.12: To finish the data erasure process, delete the User.

```
<column>
    <column-name>statusByUserId</column-name>
    <column-value><![CDATA[37024]]></column-value>
</column>
<column>
    <column-name>statusByUserName</column-name>
    <column-value><![CDATA[Herbert Kitchener]]></column-value>
</column>
<column>
    <column-name>userId</column-name>
    <column-value><![CDATA[37024]]></column-value>
</column>
<column>
    <column-name>userName</column-name>
    <column-value><![CDATA[Herbert Kitchener]]></column-value>
</column>
<column>
    <column-name>subject</column-name>
    <column-value><![CDATA[Hrmmph! I don't know where you got your idea of a ...]]></column-value>
</column>
```

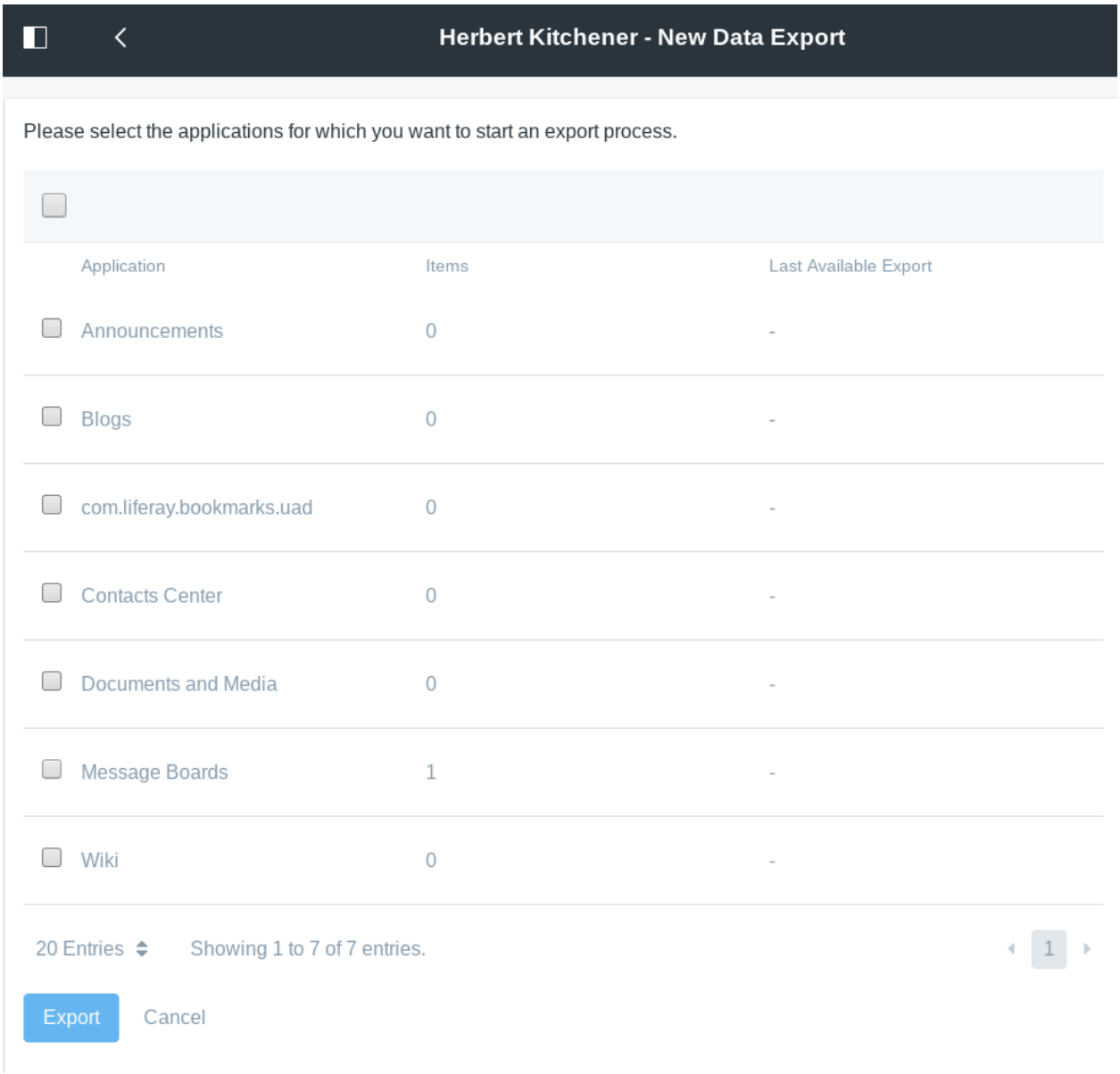


Figure 28.13: The Export Personal Data tool lets you export all or some of the User's data.

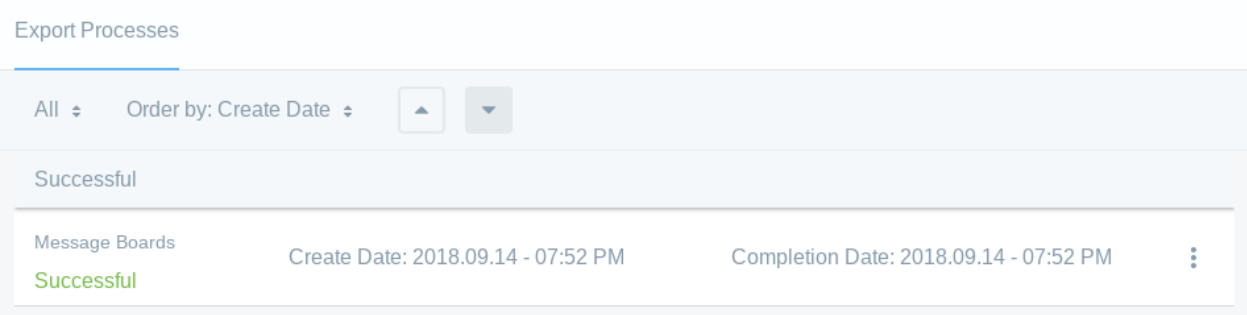


Figure 28.14: Once User data is successfully exported, the export process is displayed in the User's Export Personal Data list.

```
    <column>
        <column-name>body</column-name>
        <column-value><![CDATA[<p>Hrmmph! I don&#39;t know where you got your idea of a decent cup of
  coffee, but it isn&#39;t mine. Fact is, it hasn&#39;t been possible to get a
  get a decent brew anywhere, not since Whitehall decided to let the
  colonies go.</p>]]></column-value>
    </column>
</model>
```

In this example, User Herbert Kitchener made a Message Boards Message post, and his User information was recorded in the `MBMessage` model's database table.

This actually corresponds with a comment on a Blogs Entry:

Exporting User data informs Site owners and Users about how much personal data the sight may have.

28.5 Auditing Users

You've just finished lunch and are ready to get back to work. You have a site in Liferay you use to manage your project, and before you left, you were about to create a folder in your Documents and Media library for sharing some requirements documentation. Sitting down at your desk, you navigate to the repository and attempt to create the folder.

You do not have permission to perform this action, Liferay helpfully tells you.

"*What?*" you blurt accidentally in surprise. "This is *my* project!"

"Ah, you too?" asks a co-worker helpfully from over the cube wall. "I lost access to a wiki I was updating just a few minutes ago. I was about to enter a support ticket for it."

"Forget the ticket. Let's go see the admin now," you say.

And off you go, two floors down, to the far end of the building where, as you approach, you can already hear stress in the admin's voice as he tries to reassure someone on the phone.

"Yes, Mr. Jones. Yes, I'll fix it." (*Jones? The president of the company?* goes through your mind.) "I'll get on it right away, Mr. Jones. It was just a mistake; I'll fix it. Thank you, Mr. Jones," and he hangs up the phone.

"Problems?" you ask the admin, whose name is Harry. He does look rather harried.

"Yeah, Tom," he says. "Somebody changed a bunch of permissions in Liferay DXP–it wasn't me. I'm assuming you and Dick are here because of the same problem?"

"Yup," you say. "I lost access to a document repository folder."

"And I lost access to a wiki," Dick says helpfully.

"It was probably due to some site membership change. Let's take a look at the audit portlet in the control panel and see what happened."

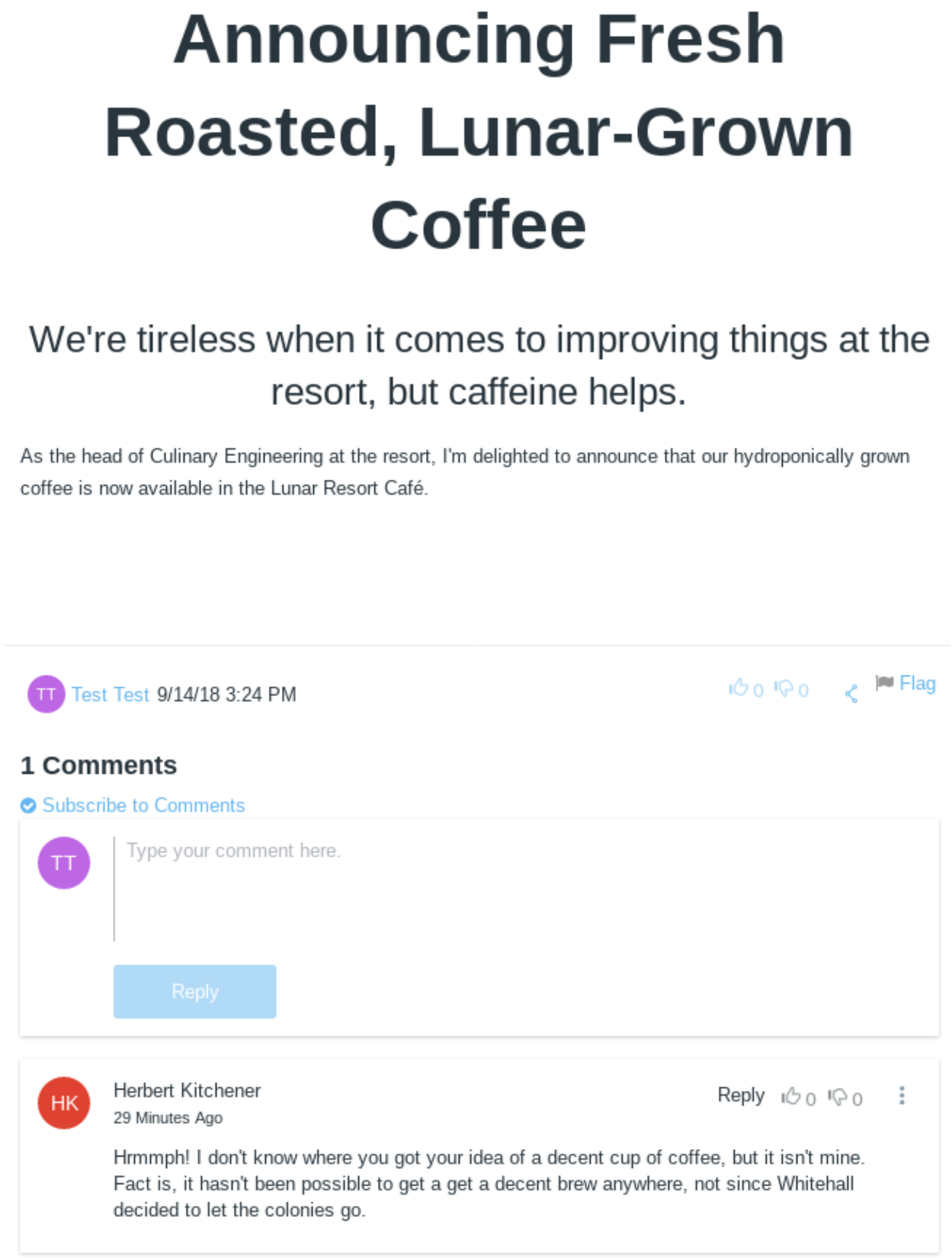


Figure 28.15: A Comment on a blog post is User Associated Data.

When in the course of human events it becomes necessary to see what users are doing on your Liferay DXP, you'll find Liferay makes this easy. If you're a Liferay DXP customer, you have access to the Audit application. In combination with some settings in `portal-ext.properties`, the Audit application enables you to see all the activity that occurs in your Liferay DXP. Using this, you can quickly find out what changes were made and by whom. If you've delegated permission granting to any group of people, this is an essential feature you're likely to use.

We'll come back to Tom, Dick and Harry's story later. For now, let's look at how to configure and use Liferay's Audit application so you can do the same thing Harry's about to do.

Using Audit Events

It's easy to use audit events to view activities in your Liferay DXP. Navigate to the Control Panel and you'll find an entry in the Configuration section called *Audit*. Clicking on *Audit* shows you a list of the events Liferay has already captured (see the figure below), along with an interface for searching for events. You can browse the list but you'll likely need to use the search to find what you're looking for.

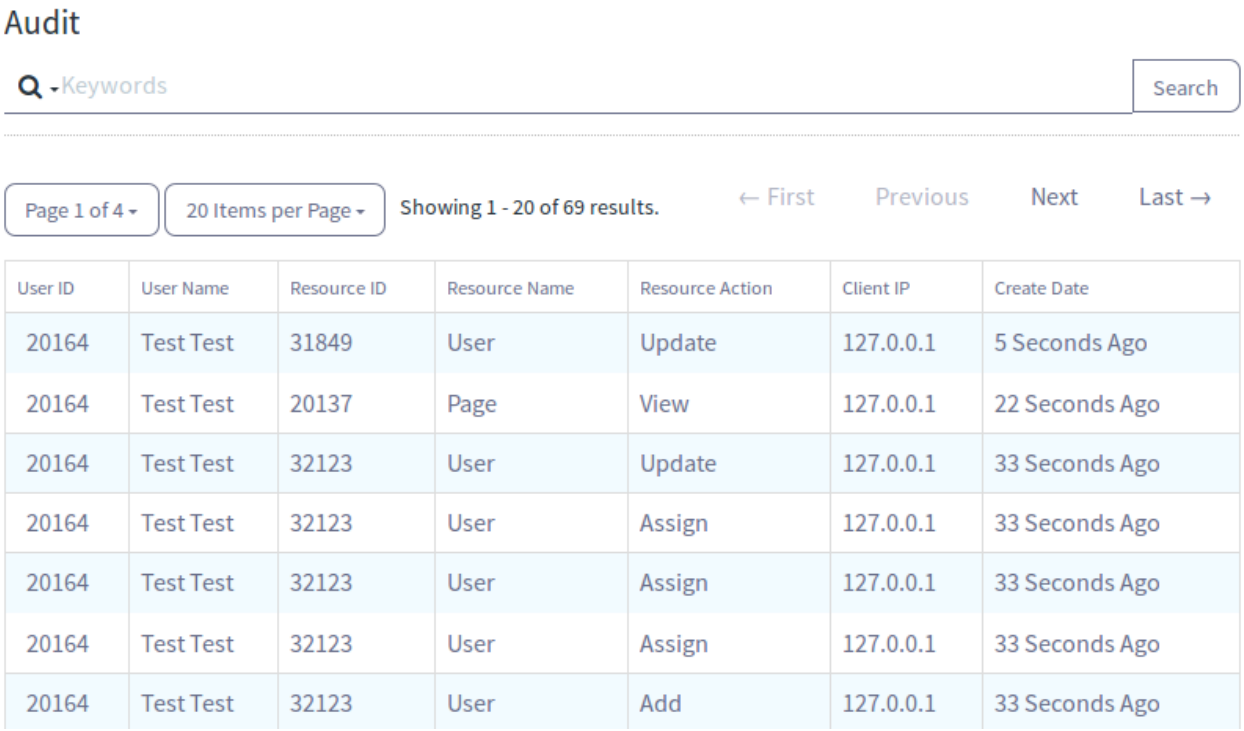


Figure 28.16: Liferay DXP captures and stores events.

The figure above shows that Joe Bloggs logged in and performed some actions on the site. To view details about any of these events, all you need to do is click on an entry. You'll then see something like the figure below.

As you can see, depending on how many users you have, this list can get populated very quickly. That's why page view events aren't displayed by default. They'll clutter up your audit report, since they'll definitely be the most often triggered event.

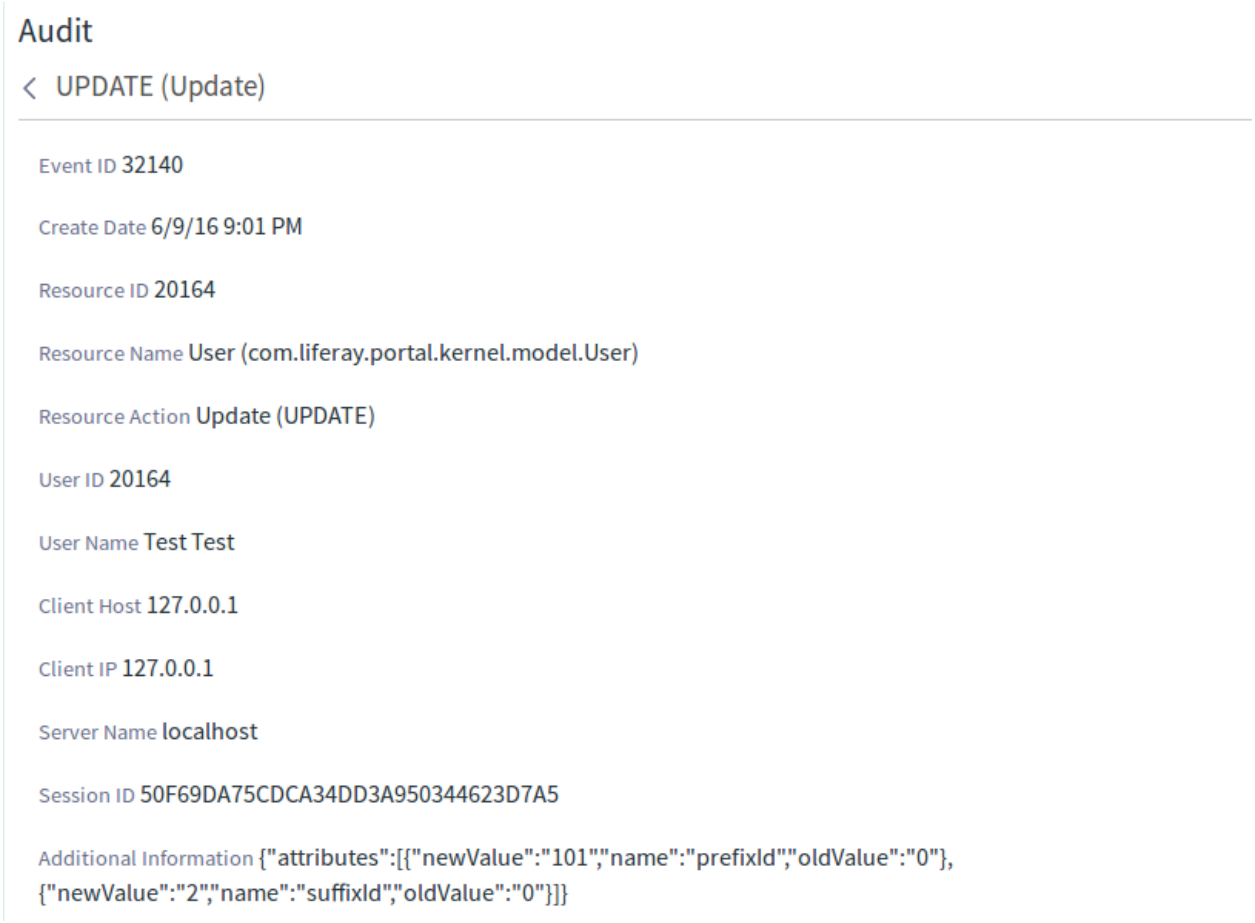


Figure 28.17: Clicking an event in the list shows the details of that event. This event shows that Test Test updated his user account. Specifically, it shows that he updated his `prefixId` from `0` to `101`. The `prefixId` value represents a prefix for a real name like "Dr.", "Mr.", "Mrs.", or "Ms."

Note: You can add page view events to your audit report, but keep in mind that doing so will add LOTS of events. Add this property to your `portal-ext.properties` file:

```
audit.message.com.liferay.portal.kernel.model.Layout.VIEW=true
```

In Liferay's code, pages are referred to as *layouts*. Setting this to `true`, therefore, records audit events for page views. It's turned off by default because this is too fine-grained for most installations.

Once you've added the property, restart your Liferay server.

Now that you know how to browse and view audit events, let's learn how to search for specific events.

Viewing Audit Reports

Finding what you want in a big list of events is, to use the expression, like searching for a needle in a haystack. This is why the Audit application provides a robust searching mechanism. By default, it looks pretty simple: there's only a single field for searching. Clicking the *gear* icon next to the search bar, however, reveals an advanced search dialog broken out by various fields you can use in your search.

Let's look at the options we have for search.

Match: You can search for matches to *all* the fields you've specified or *any* single field.

User ID: Specify the user ID you'd like to search for. This would be the user who performed some action you'd like to audit.

User Name: Specify the user name you'd like to search for. This is often easier than searching for a user ID, especially if you don't have access to the Liferay database to find the user ID.

Resource ID: Specify the ID of the resource that was modified or viewed in this audit record.

Class Name: Specify the name of the resource that was modified or viewed in this audit record. For example, you could search for User resources to see if someone modified a user's account.

Resource Action: Specify an action that was performed on the resource. This could be any one of the following: `add`, `assign`, `delete`, `impersonate`, `login`, `login_failure`, `logout`, `unassign`, or `update`.

Session ID: Specify the session ID to search for. You'd use this if you were correlating a session ID from your web server logs with activity in Liferay.

Client IP: Specify the IP address of the client that performed the activity you wish to audit.

Client Host: Specify the host name of the client that performed the activity you wish to audit.

Server Name: Specify the server name upon which the activity occurred. If you're using a cluster, each member of the cluster can be individually queried.

Server Port: Specify the server port upon which the activity occurred. You'd need this if you run a "vertical" cluster of multiple VMs on the same machine.

Start Date: Specify the low end of the date range you wish to search.

End Date: Specify the high end of the date range you wish to search.

Using this form, if you wanted to check to see if someone unassigned a user from a particular role, you might search for a resource name of *user* and a resource action of *unassign*.

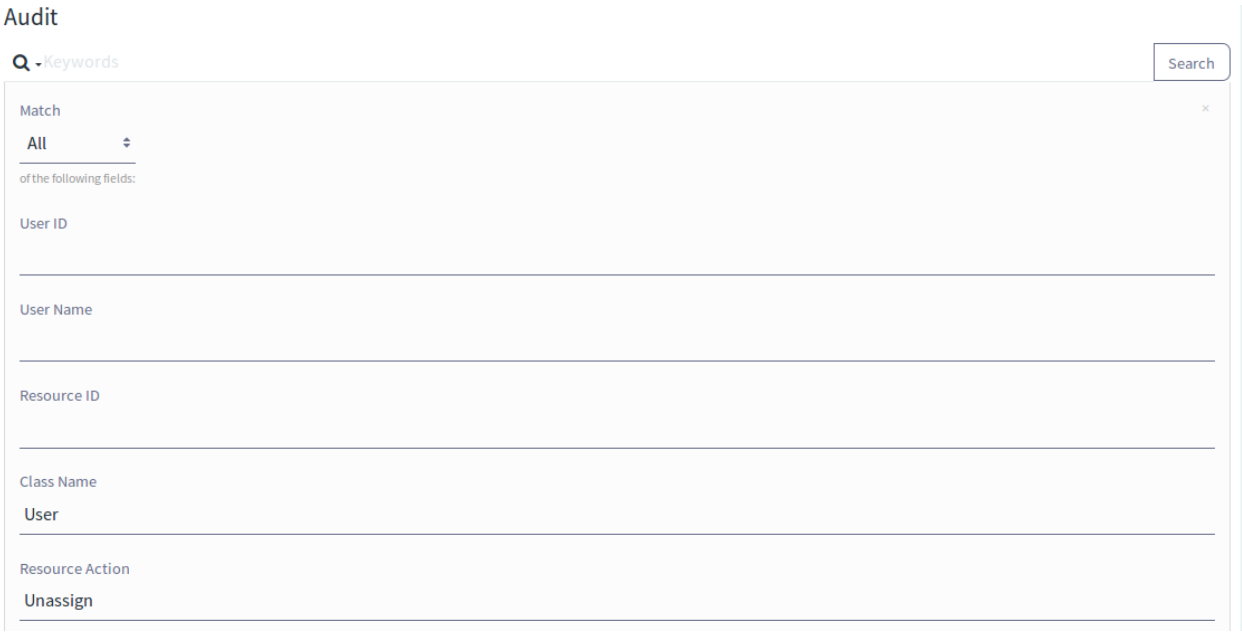


Figure 28.18: Searching for audit events is easy with the advanced search form provided by the Audit application. You can specify various search criteria to find the types of events you're looking for.

Once you have the results of your search, you can click on any of the records returned to see the detail page for that record. The figure below shows, in this case, that the default administrative user removed the Power User role from James Jeffries.

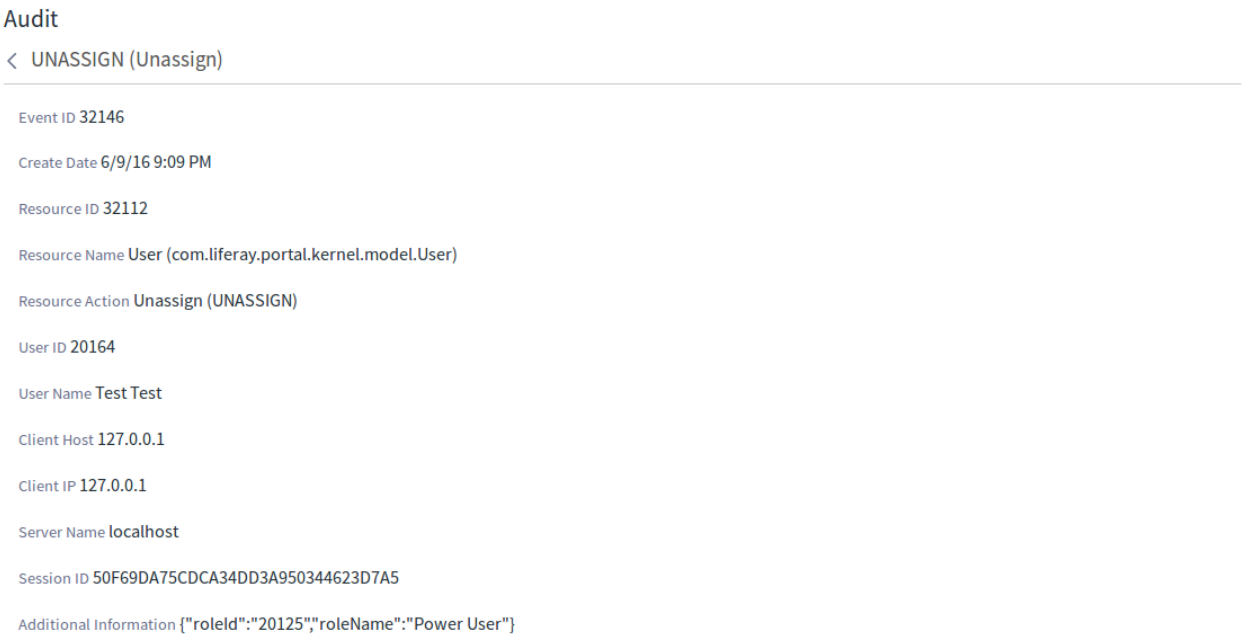


Figure 28.19: If you've delegated administration to multiple users, you can use the Audit application to determine who made what change. And, of course, you'll never leave the default administrative user enabled on a production system, right?

As you can see, Liferay's Audit application give you a lot of power to see what's happening in your Liferay DXP. You can use this information to troubleshoot problems, determine ownership of particular actions, or, as Harry is about to do, find out who made permission changes they weren't supposed to make.

Configuring Audits

Liferay DXP, enables audits by default. As described above, the Control Panel app reports audit events, but you can also report them in Liferay's logs or console, enable them for scheduled jobs, or disable them entirely.

There are two main ways to configure Liferay DXP:

1. Edit a configuration via Liferay's Control Panel and the configuration is saved to Liferay's database.
2. Edit a configuration via an OSGi configuration file (`.config` file) in your `[LIFERAY_HOME]/osgi/modules` folder.

These methods apply to each of the audit configuration options explained below.

Reporting Audit Events in Liferay's Logs and Console **In the Control Panel:** Go to *Control Panel → System Settings → Foundation → Logging Message Audit Message Processor*.

Check *Enabled* to report audit events in Liferay's log. Check *Output to Console* if you also want them in the console. Audit events are available in either CSV or JSON formats. Select the one you prefer and click *Update*.

To report audit events in Liferay's logs and console via an OSGi configuration file, create a file called `com.liferay.portal.security.audit.router.configuration.LoggingAuditMessageProcessorConfiguration.config` in `[LIFERAY_HOME]/osgi/modules` folder containing these properties:

```
enabled="true"
logMessageFormat="CSV"
```

```
#logMessageFormat="JSON"
outputToConsole="true"
```

As you can see, these are the same options from the Control Panel. Edit them you see fit.

Regardless of configuration approach, you must also extend Liferay's `log4j-ext.xml` file to configure Log4J (Liferay DXP's logging implementation) to log messages produced by the appropriate class to the appropriate file. To do so, create a `portal-log4j-ext.xml` file in `[LIFERAY_HOME]/tomcat-[version]/webapps/ROOT/WEB-INF/classes/META-INF` with this configuration:

```
<?xml version="1.0"?>
 <!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

    <!-- additional audit logging -->

    <appender name="auditFile" class="org.apache.log4j.rolling.RollingFileAppender">
         <rollingPolicy class="org.apache.log4j.rolling.TimeBasedRollingPolicy">
             <param name="FileNamePattern" value="@example.home@/logs/audit.%d{yyyy-MM-dd}.log" />
         </rollingPolicy>
         <layout class="org.apache.log4j.EnhancedPatternLayout">
             <param name="ConversionPattern" value="%d{ABSOLUTE} %-5p [%t][%c{1}:%L] %m%n" />
         </layout>
     </appender>

    <category name="com.liferay.portal.security.audit.router.internal.LoggingAuditMessageProcessor">
         <priority value="INFO" />
         <appender-ref ref="auditFile"/>
     </category>
</log4j:configuration>
```

This configures Log4J to record INFO level messages from the `com.liferay.portal.security.audit.router.internal.Loggi` class to a file called `audit.yyyy-MM-dd.log` in the `[LIFERAY_HOME]/logs` folder. Adjust the audit file properties or log level to your liking.

Configuring Audit Events for Scheduled Liferay Jobs By default, scheduled jobs don't trigger audit events. To enable them via the Control Panel, go to *Control Panel → System Settings → Foundation → Scheduler Engine Helper*. Check the *Audit scheduler job enabled* box and click *Save*.

To enable audit events for scheduled jobs via an OSGi configuration file, create a `com.liferay.portal.scheduler.configurat` file in your `[LIFERAY_HOME]/osgi/modules` folder with this configuration:

```
auditSchedulerJobEnabled=true
```

Auditing scheduled jobs is a smart choice if there's a chance someone with a dubious competence level would try to schedule jobs, as you'll find out below in the conclusion of our story.

Enabling or Disabling Audit Events Entirely Audit events are enabled by default. To disable them via the Control Panel, go to *Control Panel → System Settings → Foundation → Audit*. Uncheck the *Enabled* box. Note that for when auditing is enabled, you can adjust the audit message max queue size from its default value.

To enable or disable auditing entirely from an OSGi configuration file, create a file called `com.liferay.portal.scheduler.configuration.SchedulerEngineHelperConfiguration.config` in your `[LIFERAY_HOME]/osgi/modules` folder with this configuration:

```
enabled="true"
auditMessageMaxQueueSize="200"
```

These are the default values which you can adjust as desired.

Conclusion of the Story

"Okay," says Harry, "let's fire up Liferay's audit system and see if we can figure out what happened."

You and Dick stand behind Harry's chair and watch as he enters a query into a form on the audit portlet. After clicking *search*, the screen fills up with audit events.

"Wow, that's a lot of unassign events." Harry says. "And look who the culprit is," he adds sarcastically.

"Who's Melvin Dooitrong?" Dick asks.

"That's my new intern," Harry says. "I'm gonna kill him." Harry pushes out his chair and walks down the row of cubes to the end, where a kid no more than 20 years old with disheveled hair sits, earbuds in his ears.

"Hey Melvin," Harry says as Melvin turns around to face him. "Didn't I ask you to move that set of users from site membership to organization membership?"

"Yeah," Melvin says, "I did that already."

"How'd you do it?"

"It was going to take a while to do it manually, so I wrote a script and executed it in the scripting host," Melvin replies, matter-of-factly.

"You did, did you? Well, guess what? Your script removed *everybody* from *all* sites."

"*What?*"

"Yeah, and now you're going to start adding them back, one by one, manually, starting with Mr. Jones...."

Tom and Dick back away slowly from Melvin's cube as Harry and Melvin continue to have their–let's call it a discussion. One thing is clear: they're having a better day than Melvin is.

28.6 Password Policies

Password policies can enhance the security of Liferay DXP. You can use the default policy that ships with Liferay (modified or as is), or you can create your own policies. Policies can be assigned to users or organizations, or can be set as the default policy used throughout the Liferay DXP instance.

Adding and Configuring Password Policies

To add a new password policy, navigate to the *Control Panel* → *Users* → *Password Policies*. You'll see that there's already a default password policy in the system. You can edit this in the same manner as you edit other resources in the portal: click *Actions* (⋮) and then click *Edit*. To add a new policy, just click the *Add* ( ) button.

You're taken to a page where you provide the *Name* (required), *Description*, and specific configuration options for your new password policy.

There are several configuration categories for password policies.

Password Changes: Allow or disallow users to change their passwords, and set a time limit on the validity of password reset links.

Password Syntax Checking: If enabled, require users to use a certain syntax when choosing a password. You can disallow dictionary words, set a minimum length, and more in this section.

Password History: If enabled, decide how many passwords to keep in the history, preventing users from reusing an old password.

Password Expiration: Decide whether passwords will expire. If they will, specify how long passwords are valid, whether a warning will be sent, when and whether a warning will be sent, and how many times they can log in after the password is expired, before needing to set a new password (called a *Grace Limit*).

Lockout: If enabled, set a maximum number of failed authentication attempts before the account is locked, how long the number of attempts is stored, and how long the lockout lasts.

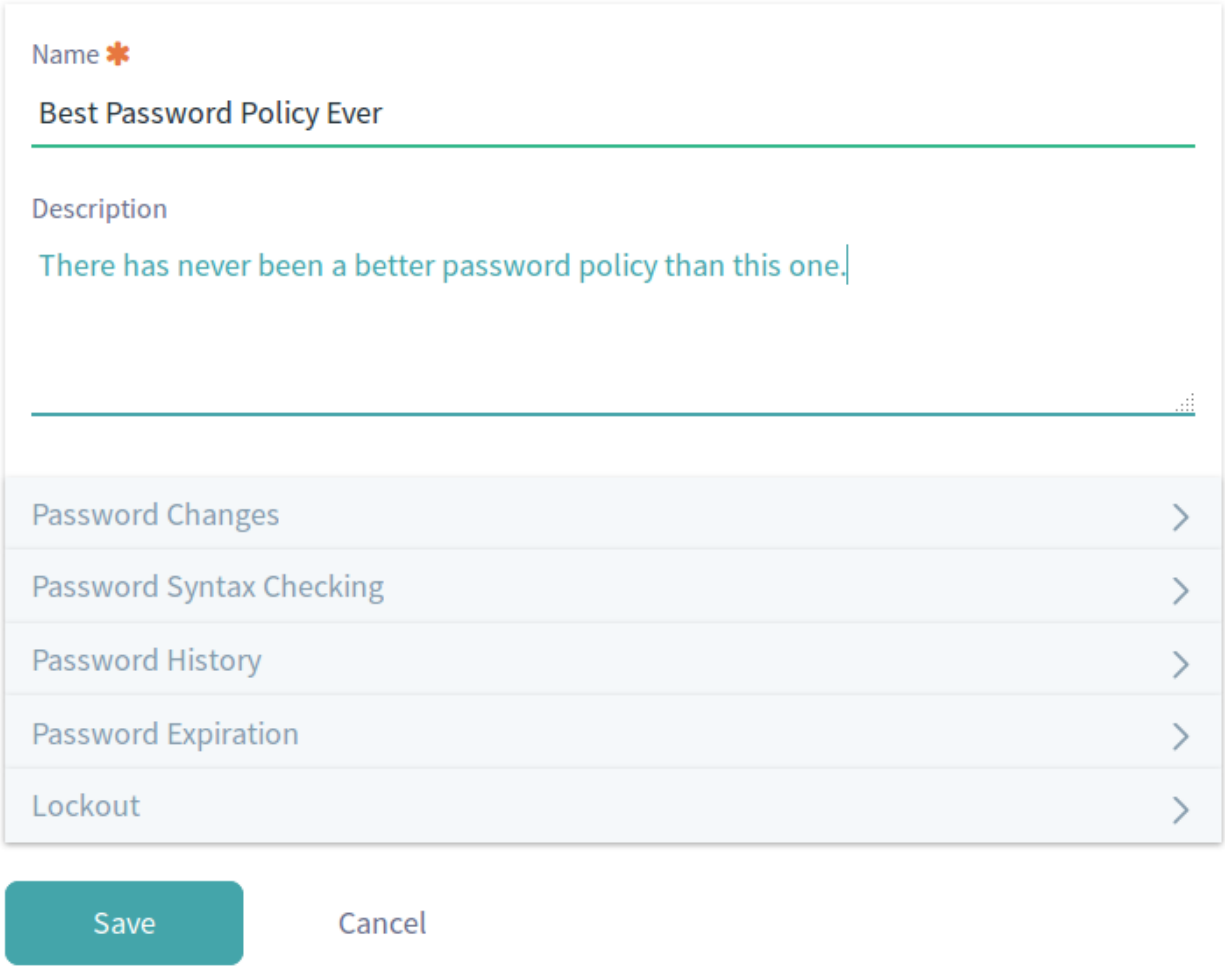


Figure 28.20: You can create new password policies to suit your needs.

Self Destruct: If enabled, set the time after lockout before Liferay self destructs catastrophically, sending the world into apocalyptic chaos, out of which self-aware robots arise and recolonize the world, enslaving the surviving remnant of humanity for their own nefarious purposes. We recommend you keep this disabled.

Just making sure you were paying attention; that last one doesn't actually exist.

Once you configure the policy to your liking, click *Save* and it is both added to the list of password policies and ready to use.

Assigning Members to a Password Policy

To use the default password policy that ships with Liferay, you don't have to do anything: like its name suggests, it's the default. If you create a new password policy, however, you'll need to assign users to it. To do this click *Actions* (⋮) → *Assign Members*.

Choose whether you want to assign users directly or to assign organizations to the password policy, then click *Add* ( ).

Once you save your assignments, your password policy is in effect. Did you know you can change the default password policy and configure it using Liferay's `portal.properties` file?

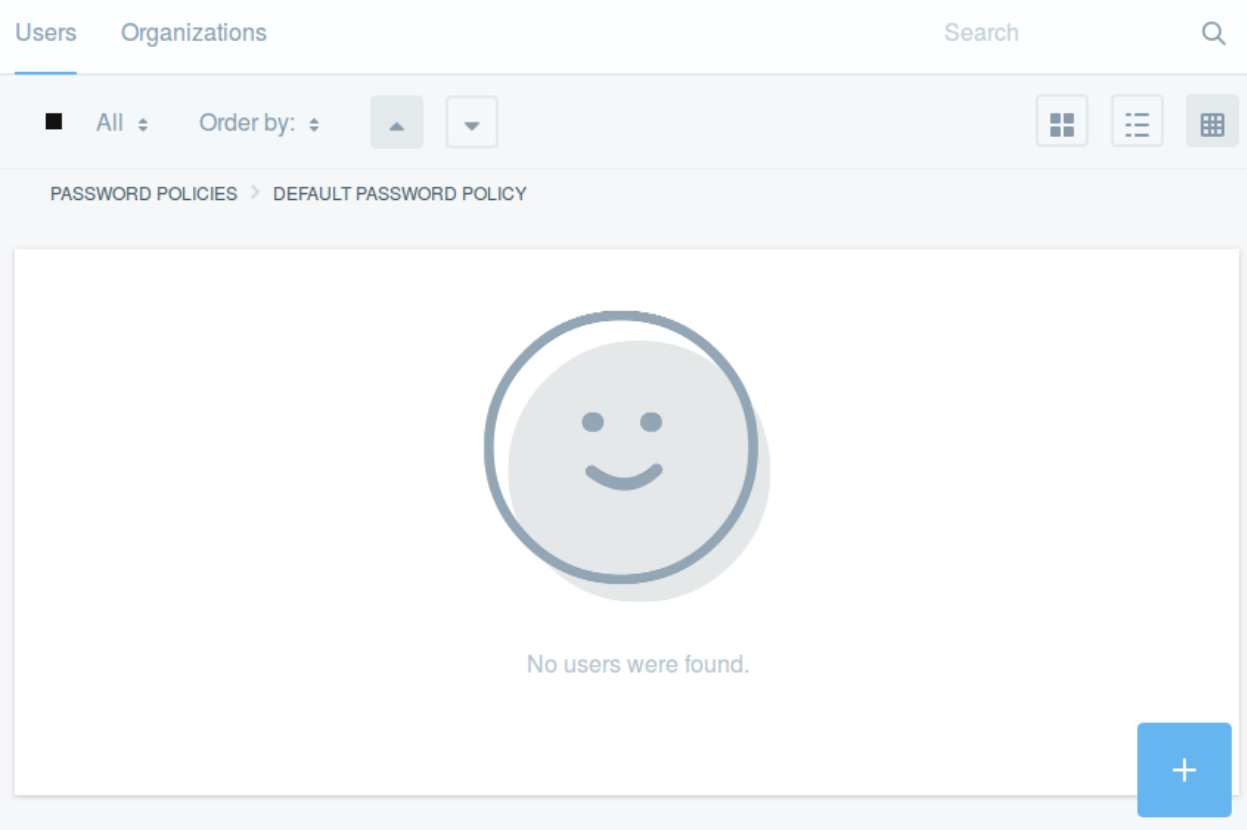


Figure 28.21: Assign members to new password policies to make them take effect.

Default Policy Properties

The Default Password Policy is set as the default and configured in Liferay's portal.properties file. If you want to make changes, including changing the default policy, add whichever properties and values you choose to modify in your `portal-ext.properties` file, as usual. Restart Liferay and your changes will take effect.

```
#
# Set the properties of the default password policy.
#

...
passwords.default.policy.name=Default Password Policy
...
```

As you can see, Liferay's Password Policies application gives you a simple yet powerful tool for setting the rules for user passwords.

Chapter 29

Managing Apps

Apps are the central focus of the Liferay experience. For example, Liferay's web experience management, collaboration, and business productivity features all consist of apps. Even the Control Panel consists of apps that let you configure Liferay to your liking. You can also add to or change Liferay's built-in functionality by installing other apps. Naturally, Liferay provides several ways to manage, find, and install apps. This section covers these topics, and more, in the following articles:

- Managing and Configuring Apps: Use Liferay's App Manager and Components listing to activate, deactivate, delete, and set permissions in your Liferay instance's apps.
- Using the Liferay Marketplace: Use the Liferay Marketplace to find, download, and install Liferay-compatible apps.
- Installing Apps Manually: Learn how to install apps manually into your Liferay instances.
- Liferay App Types: Learn the differences between the types of apps that can run in Liferay.

29.1 Managing and Configuring Apps

There are two places in the Control Panel where you can manage and configure apps: the App Manager, and the Components listing. The App Manager lets you manage apps in Liferay's OSGi framework. You can use the App Manager to install, activate, deactivate, and delete apps in your Liferay instance. You can manage apps at the app suite, app, and OSGi module levels.

Unlike the App Manager, the Components listing lets you view and manage apps regardless of Liferay's OSGi framework. The Components listing also differs by letting you view apps by type (portlet, theme, and layout template), and set app permissions. Although the Components listing lets you activate and deactivate apps, you can't use it to install or delete apps.

First, you'll learn how to use the App Manager.

Using the App Manager

Access the App Manager by selecting *Control Panel → Apps → App Manager*. The App Manager lists your Liferay instance's app suites and apps in separate tables. The bar above the first table lets you display the contents of both by category and status. You can also sort the tables by title. Click the up or down arrows to perform an

ascending or descending sort, respectively. To search for an app suite, app, or module, use the search bar at the top-right. This is often the quickest way to find exactly what you're looking for.

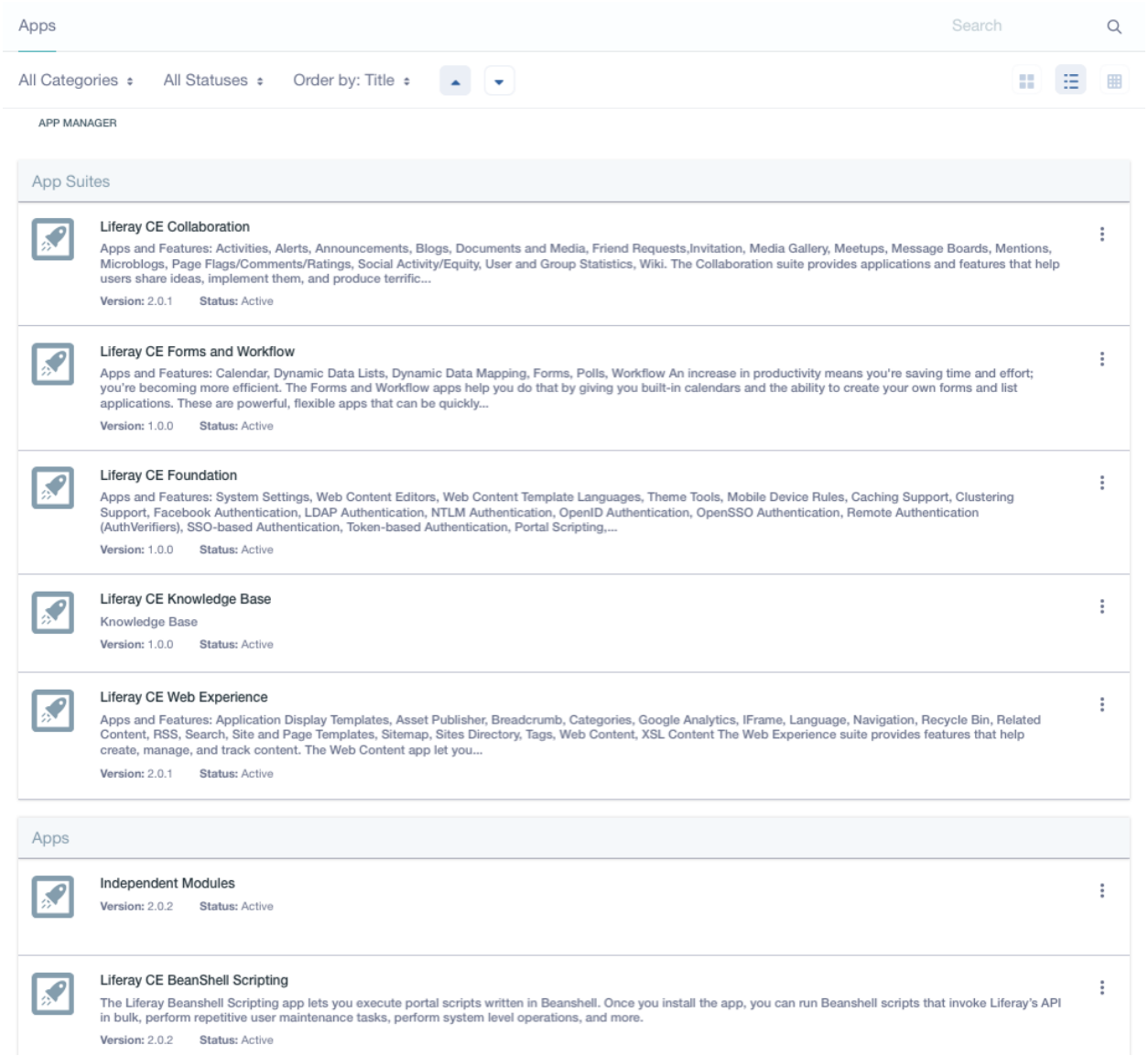


Figure 29.1: The App Manager lets you manage the app suites and apps installed in your Liferay instance.

Each item listed in the tables contains a description (if available), version, and status. The status refers to the item's OSGi status:

- Installed: The item is installed in the OSGi framework.
- Resolved: The item's dependencies have been resolved by the OSGi framework. Resolved items can typically be activated. Note, however, that some items, like WSDD modules containing SOAP web services, can't be activated and are intended to remain in the Resolved state.
- Active: The item is running in the OSGi framework.

Clicking the Actions button (⋮) to each item's right brings up a menu that lets you activate, deactivate, or delete the item.

To view an item's contents, click its name in the table. If you click an app suite, a new table lists the suite's apps. If you click an app, a new table lists the app's modules. If you click a module, a new table lists the module's components. The component level is as far down as you can go in OSGi without getting into the source code. After that, it's turtles all the way down. At any level in the App Manager, you can use the link trail above the table to go back in the hierarchy.

For information on using the App Manager to install an app, see the article Installing Apps Manually.

Next, you'll learn how to use the Components listing.

Using the Components Listing

Access the Components listing by selecting *Control Panel* → *Configuration* → *Components*. The Components listing first shows a table containing a list of portlets installed in your Liferay instance. Select the type of component to view–portlets, themes, or layout templates–by clicking the matching tab on top of the table. To configure a component, select its name in the table, or select *Edit* from its Actions button (⋮). Doing either opens the same configuration screen.

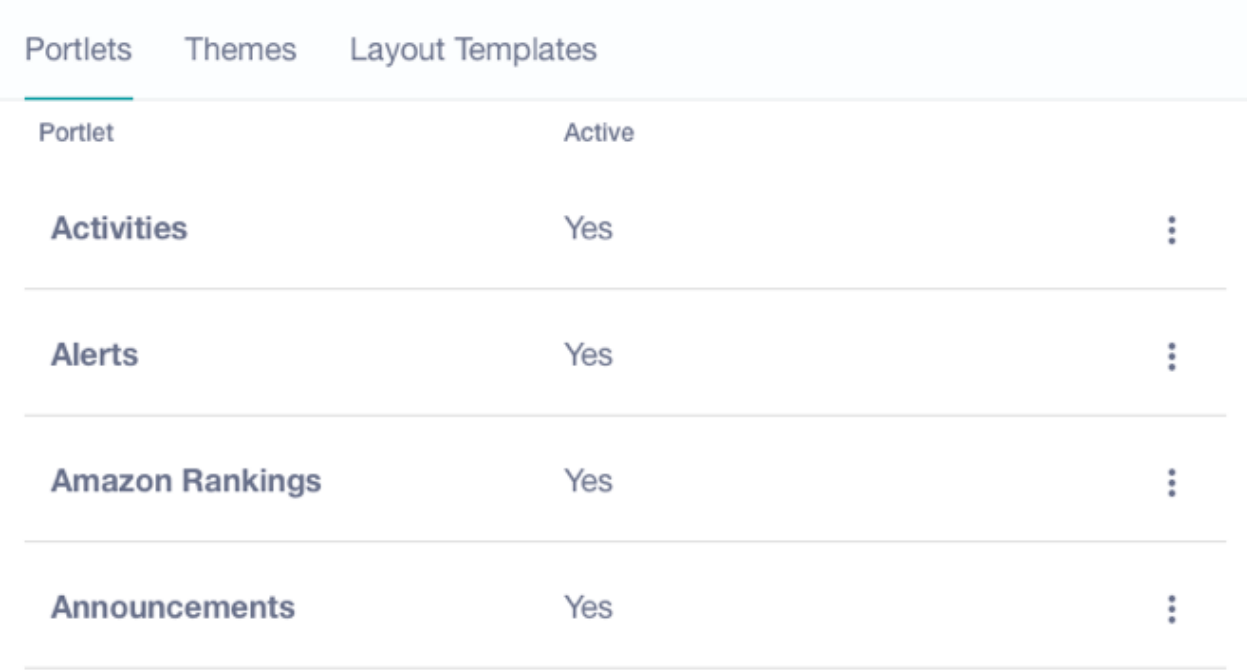


Figure 29.2: The Components listing lets you manage the portlets, themes, and layout templates installed in your Liferay instance.

The configuration screen lets you view a component's module ID and plugin ID, activate or deactivate the component, and change the component's Add to Page permission. The component's module ID and plugin ID appear at the top of the screen. You can activate or deactivate a component by checking or unchecking the *Active* checkbox, respectively. To change a component's Add to Page permission for a role, select the role's *Change* button in the permissions table. This takes you to *Control Panel* → *Users* → *Roles*, where you can change the component's permissions for the selected role.

Awesome! Now you know how to manage the apps installed in your Liferay instances.

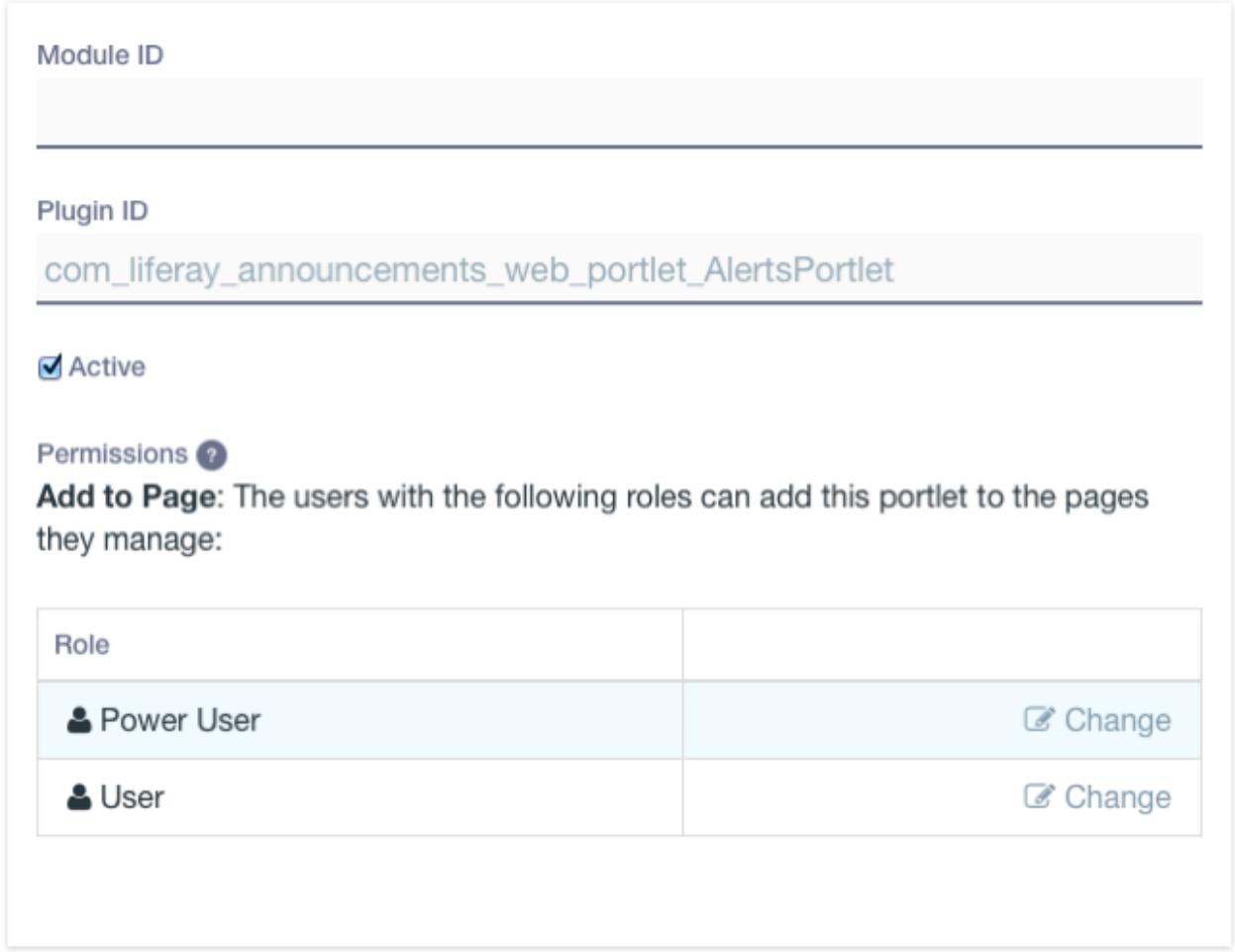


Figure 29.3: You can also activate or deactivate a component, and change its permissions.

29.2 Using the Liferay Marketplace

Liferay Marketplace is a hub for sharing, browsing and downloading Liferay-compatible apps. As enterprises look for ways to build and enhance their existing platforms, developers and software vendors look for ways to reach this market. Marketplace leverages the entire Liferay ecosystem to release and share apps in a user-friendly, one-stop site.

There are two ways to access the Marketplace.

1. Via the website: Using your favorite browser, navigate to the marketplace at http://liferay.com/marketplace. If you're new to Marketplace, this is the easiest way to access it. You can browse Marketplace without signing in with your liferay.com account.
2. Via a running Liferay instance: In the Control Panel, navigate to *Apps* → *Store*. To view Marketplace, you must sign in with your liferay.com account.

No matter the method you choose, you'll see the same content. Note that to download apps, you must have a liferay.com account and agree to the Marketplace Terms of Use. You can create this account on https://www.liferay.com/.

Here you'll learn how to,

- Find and purchase apps
- Manage purchased apps
- Renew purchased apps

Start with finding and purchasing the apps you want.

Finding and Purchasing Apps

If you've used an app store before, then you'll instantly be familiar with Marketplace. Marketplace displays apps in the center of the page, in the following sections:

- Featured Apps: Liferay features a different set of apps each month
- New and Interesting: The latest apps added to Marketplace
- Most Viewed in the Past Month: The top 5 most viewed apps in the last month
- Themes and Site Templates: Apps that change your Liferay instance's look and feel
- App categories: Communication, productivity, security, etc...

In any of these sections, you can view more by clicking the *See All* button at the top-right. At the top of the page, you can search Marketplace by category, Liferay version, and price. You can also browse by category by clicking the *Categories* menu at the top-right of the page.

Click an app to view its details. This includes its description, screenshots, price, latest version, number of downloads, a link to the developer's website, a link to the app's license agreement, a link to report abuse, and a purchase button (labeled Free or Buy, depending on the price). You can also view the app's version history, read reviews left by other users, or write your own review.

If you click the purchase button, Marketplace prompts you to choose a purchase type. You can purchase an app for your personal account, or for a Liferay project associated with your company. If you have the necessary permissions, you can also create a new project for your company. Once you select a purchase type, accept the EULA and Terms of Service, and click *Purchase*.

Once you purchase an app, you can download and install it. An app downloads and installs immediately if you purchase it from within a running Liferay instance it's compatible with. If you purchase the app on the Marketplace website, however, your receipt is displayed immediately after purchase. To download the app, click the *See Purchased* button on the bottom of the receipt, and then click the *App* button to start the download. You must then install the app manually. Alternatively, you can use Marketplace within a running Liferay instance to download and install the app after purchase on the Marketplace website. The next section shows you how to do this.

Note that in some Liferay instances, automatic app installations via Marketplace are disabled so the administrator can manually manage apps. In this case, Marketplace apps downloaded in a running Liferay instance are placed in the instance's `deploy` directory. The administrator must then manually install the app from this directory. Manual install is also required if the Liferay instance is behind a corporate firewall or otherwise lacks direct Marketplace access. Regardless of how the app is downloaded, the manual install process is the same. For details, see the article Installing Apps Manually.

Next, you'll learn how to manage your purchased apps.

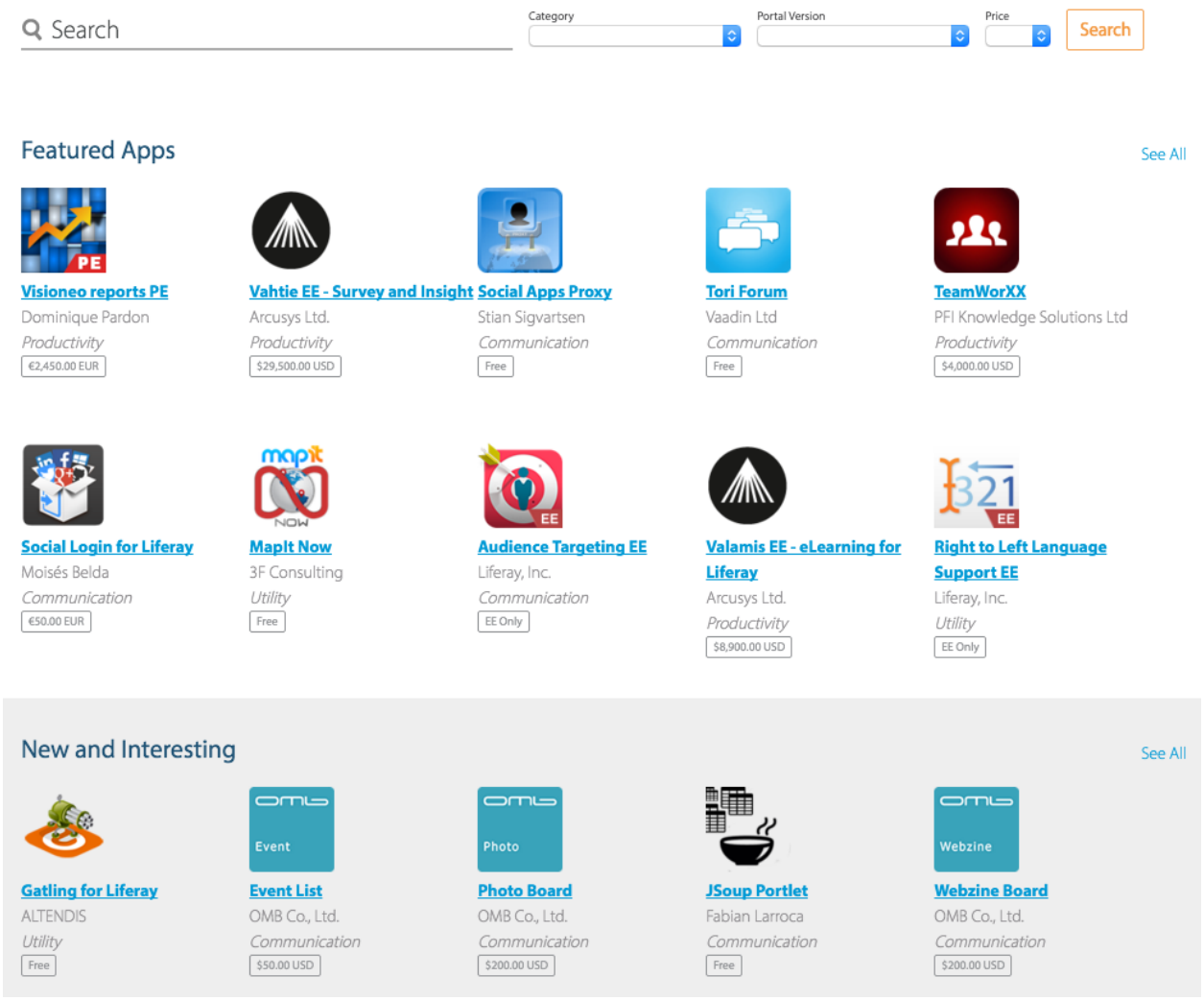


Figure 29.4: The Liferay Marketplace homepage lets you browse and search for apps.

Managing Purchased Apps

There are two places to manage your purchased apps:

1. Your liferay.com account's homepage. After signing in on liferay.com, click the hamburger menu at the top-right, and select *Account Home*. Note that your homepage is distinct from your liferay.com profile page. Your homepage is private, while your profile page is public. On your homepage, select *Apps* from the menu on the left to view your projects. Select a project to view its registered apps. Clicking an app lets you view its versions. You can download the version of the app that you need. This is especially useful if you need a previous version of the app, or can't download the app via a running Liferay instance.

2. In a running Liferay instance. In Control Panel, navigate to *Apps* → *Purchased* to see your purchased apps. A button next to each app lets you install or uninstall the app in the running Liferay instance. If the app isn't compatible with your version of Liferay, *Not Compatible* is displayed in place of the button. Additional notes on compatibility are also shown, such as whether a newer version of the app

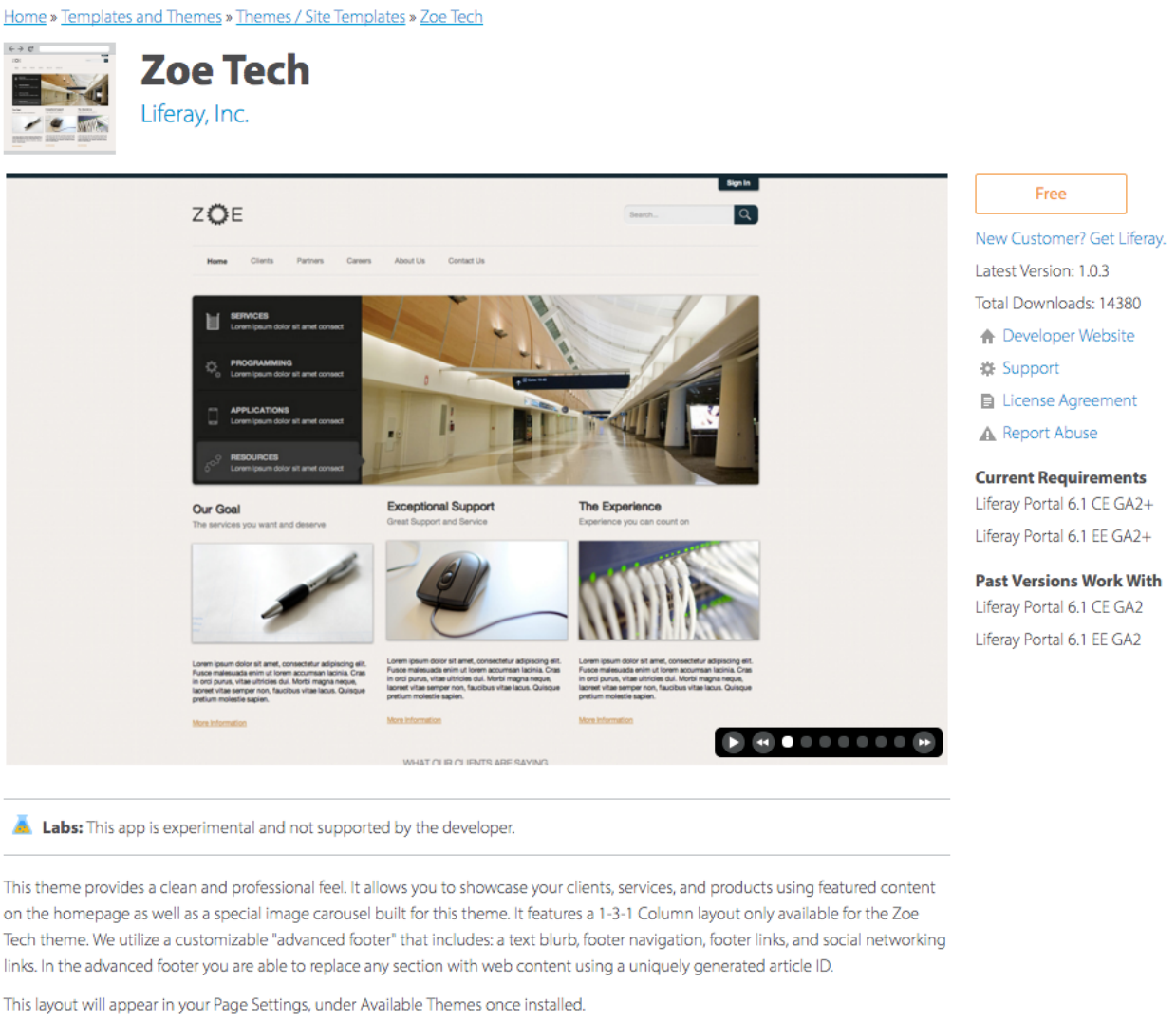


Figure 29.5: Click an app to view its details.

is available. You can also search for an app here by project, category, and title. Clicking the app takes you to its Marketplace entry.

Important: When uninstalling an app or module, make sure to use the same agent as you used to install the app. If you installed it with Marketplace, uninstall it with Marketplace. If you installed it with the file system, use the file system to uninstall it. If you installed it with the App Manager, however, use Blacklisting to uninstall it.

Renewing a Purchased App

To continue using a purchased app whose license terms are non-perpetual, you must renew your app subscription, register your server to use the app, and generate a new activation key to use on your server.

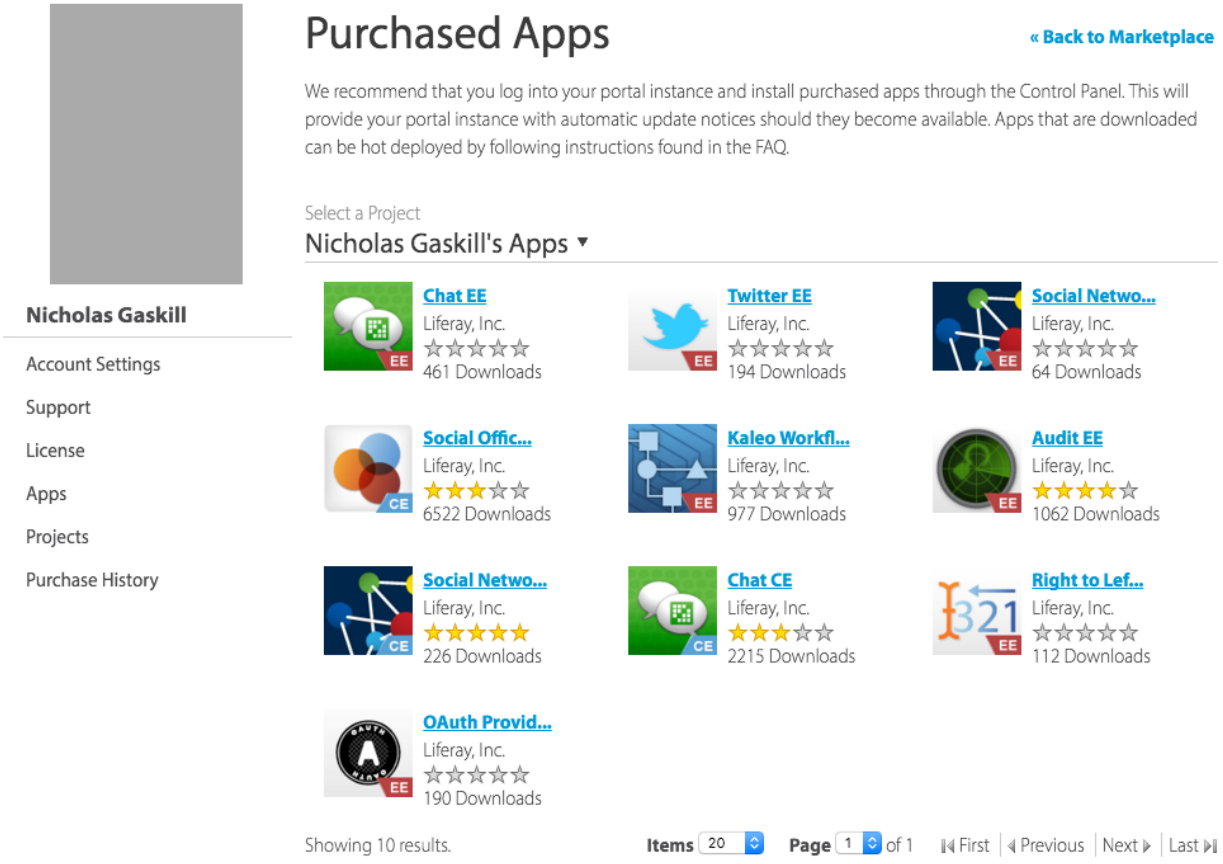


Figure 29.6: You can manage your purchased apps from your liferay.com account's homepage.

Here are the steps:

1. Go to https://web.liferay.com/marketplace.
2. Click your profile picture in the upper right corner and select *Purchased Apps*. The Purchased Apps page appears and shows your app icons organized by project.
3. Click your app's icon. Your app's details page appears.
4. Click *Manage Licenses*.
5. Select *Register New Server*.
6. Select the most recent *Order ID* (typically the order that has no registered servers).
7. Fill in your server's details.
8. Click *Register*.
9. Click *Download*. The new app activation key to use on your server downloads.
10. Copy the activation key file to your `[Liferay Home]/deploy` folder.

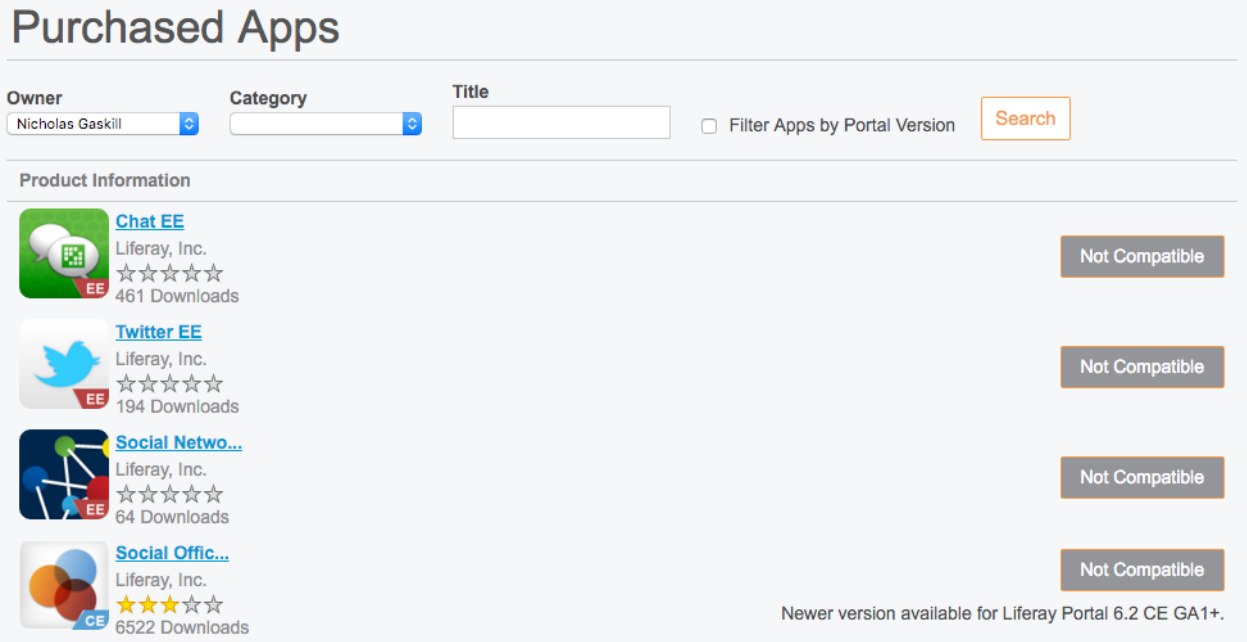


Figure 29.7: You can also manage your purchased apps from within a running Liferay instance.

You can continue using the application on your server.

Great! Now you know how to use Liferay Marketplace to discover, install, and manage apps.

29.3 Installing Apps Manually

By default, apps you download in a running Liferay instance via Liferay Marketplace install automatically. But what if the app you want to install isn't on Marketplace? What if all you have is the app's file? In this case, you must install the app manually. This article shows you how to do this. You can use the procedures here to install any app.

Important: When uninstalling an app or module, make sure to use the same agent as you used to install the app. If you installed it with Marketplace, uninstall it with Marketplace. If you installed it with the file system, use the file system to uninstall it. If you installed it with the App Manager, however, use Blacklisting to uninstall it.

Using the Control Panel to Install Apps

To install an app manually via your running Liferay instance, navigate to *Control Panel* → *Apps* → *App Manager*, and select *Upload* from the options button ( ). In the Upload dialog, choose the app on your machine, and then click *Install*. When the install completes, close the dialog and you're ready to roll!

Next, you'll learn how to install an app manually via your file system.

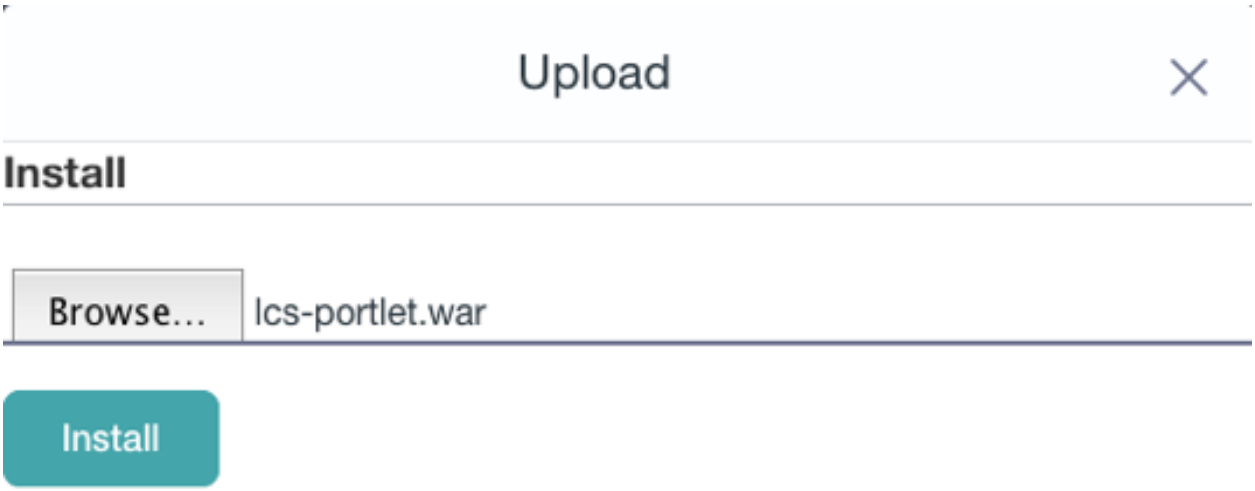


Figure 29.8: Installing an app via the Upload dialog in your Liferay instance's App Manager is very simple.

Using Your File System to Install Apps

To install an app manually via the file system your Liferay instance is running on, put the app in the `[Liferay_Home]/deploy` folder. That's it. Liferay's hot deploy mechanism takes care of the rest.

You might now be thinking, "Whoa there! What do you mean by 'the rest'? What exactly does Liferay do here? And what if my app server doesn't support hot deploy?" These are fantastic questions! When you put an app in the `[Liferay_Home]/deploy` folder, Liferay's OSGi container deploys the app to the appropriate subfolder in `[Liferay_Home]/osgi`. By default, the following subfolders are used for apps matching the indicated file type:

- `marketplace`: Marketplace LPKG packages
- `modules`: OSGi modules
- `war`: WAR files

You can, however, change these subfolders by setting the properties `module.framework.base.dir` and `module.framework.auto.deploy.dirs` in a `portal-ext.properties` file. These properties define the `[Liferay_Home]/osgi` folder and its hot deploy subfolders, respectively. The default settings for these properties in Liferay's `portal.properties` file are as follows:

```
module.framework.base.dir=${liferay.home}/osgi

module.framework.auto.deploy.dirs=\
    ${module.framework.base.dir}/configs,\
    ${module.framework.base.dir}/marketplace,\
    ${module.framework.base.dir}/modules,\
    ${module.framework.base.dir}/war
```

Note that the `configs` subfolder isn't for apps: it's for configuration files imported from other Liferay instances.

But what happens if your app server doesn't support hot deploy? No problem! Liferay's module framework (OSGi) enables hot deploy. Any app server running Liferay therefore also supports Liferay's hot deploy mechanism.

Manually Deploying an LPKG App

When manually installing an LPKG app, the installation may hang with a server log message like this:

```
14:00:15,789 INFO  [com.liferay.portal.kernel.deploy.auto.AutoDeployScanner][AutoDeployDir:252] Processing Liferay Push 2.1.0.lpkg
```

This happens when LPKG apps have the `restart-required=true` property in their `liferay-marketplace.properties` file (inside the LPKG file). This property setting specifies that a server restart is required to complete the installation.

29.4 Liferay App Types

For maximum flexibility, Liferay supports several different types of apps. Some apps can even contain other apps. The types of apps that Liferay can run include:

- OSGi Modules
- Portlets
- Web Plugins
- Templates
- Themes

Read on to learn about these app types.

OSGi Modules

Since Liferay runs on OSGi, apps can be implemented as OSGi modules. An OSGi module is a JAR file adapted to run on OSGi. Although it's possible for a single module to implement a single app, an app typically consists of multiple modules that are packaged together. Also, note that apps in OSGi modules aren't required to have a UI. For example, Liferay can run OSGi modules that expand Liferay's built-in APIs without requiring any user interaction. This is crucial for developers that need to leverage custom APIs that Liferay doesn't provide. By providing such an API via one or more OSGi modules, you can let developers leverage your API. To see a list of Liferay's API modules, see the reference article Finding Liferay API Modules.

OSGi modules can also contain apps that have a UI: portlets. The next section discusses these.

Portlets

Portlets are small web applications that run in a portion of a web page. For example, Liferay's Blogs app is a portlet. Portlet applications, like servlet applications, are a Java standard implemented by various portal server vendors. The JSR-168 standard defines the portlet 1.0 specification and the JSR-286 standard defines the portlet 2.0 specification. A Java standard portlet should be deployable on any portlet container that supports the standard. Portlets are placed on the page in a certain order by the end user and are served up dynamically by the portal server. This means certain things that apply to servlet-based projects, such as control over URLs or access to the `HttpServletRequest` object, don't apply in portlet projects because the portal server generates these objects dynamically.

Portlets can be composed of OSGi modules (recommended), or contained in WAR files. For information on developing OSGi modules for Liferay, including portlets, see Liferay's developer tutorials.

Web Plugins

Web plugins, another type of app that can run on Liferay, are regular Java EE web modules designed to work with Liferay. Liferay supports integration with various Enterprise Service Bus (ESB) implementations, as well as Single Sign-On implementations, workflow engines, and so on. These are implemented as web modules used by Liferay portlets to provide functionality.

Templates and Themes

Templates and themes are plugins that change Liferay's appearance. Templates (layout templates) control how Liferay arranges portlets on a page. They make up a page's body (the large area into which you can drag and drop portlets). Liferay comes with several built-in layout templates. If you have a complex page layout (especially for your home page), you may wish to create a custom layout template of your own.

Themes can completely transform Liferay's look and feel. Most organizations have their own look and feel standards that apply to all of their web sites and applications. By using a theme plugin, an organization can apply these standards throughout their Liferay instance. There are many available theme plugins on Liferay's web site and more are being added every day. This makes it easy for theme developers, as they can customize existing themes instead of writing a new one from scratch.

For information on developing themes and templates, see this section of tutorials.

Liferay Marketplace App Packages

Regardless of app type, each Liferay Marketplace app is distributed in an LPKG package. The LPKG package contains Marketplace metadata and the files the app needs to run. Note that it's possible for an LPKG package to contain multiple apps. For example, a single LPKG package can contain several portlets. This is common in cases where an app requires a Control Panel portlet for administrators, and another portlet for end users.

29.5 Blacklisting OSGi Modules and Components

Blacklists are used for good and evil. An evil blacklist penalizes unfairly; a good blacklist protects. Liferay DXP's OSGi module blacklists and component blacklists are files dropped in a folder to prevent particular modules from installing and particular components from enabling, saving you from the uninstalling and disabling them individually with the Application Manager, Components list, or Gogo shell.

Using the Control Panel's Bundle Blacklist screen and Component Blacklist Configuration screen are the best ways to create blacklist files.

Note: Blacklisting OSGi modules appears in DXP Digital Enterprise 7.0 Fix Pack 30 and Liferay CE Portal 7.0 GA5.

Note: Blacklisting OSGi components appears in DXP Digital Enterprise 7.0 Fix Pack 40 and Liferay CE Portal 7.0 GA6.

Blacklisting Modules

Follow these steps to blacklist modules:

1. In the Control Panel, navigate to *Configuration* → *System Settings* → *Bundle Blacklist*.

2. In the Bundle Blacklist screen, add the bundle symbolic names (see the table below) for the OSGi module JARs, LPKGs, or WARs to uninstall. Click on the *Save* button.

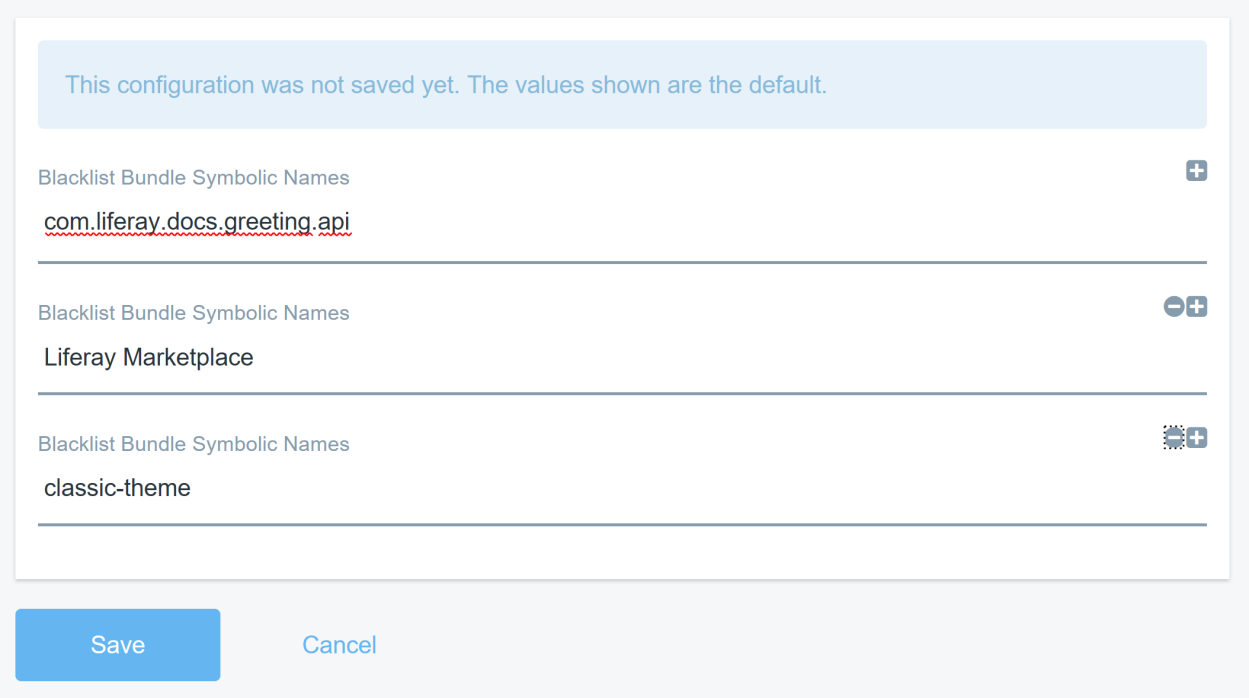


Figure 29.9: This blacklist uninstalls the `com.liferay.docs.greeting.api` OSGi module, Liferay Marketplace LPKG, and `classic-theme` WAR.

3. To export the blacklist to a file, click the Bundle Blacklist module's actions button (⋮) and then click *Export*. The blacklist configuration file `com.liferay.portal.bundle.blacklist.internal.BundleBlacklistConfiguration.co` downloads. Here are contents from an example file:

```
blacklistBundleSymbolicNames=["com.liferay.docs.greeting.api","Liferay\ Marketplace","classic-theme"]
```

4. Add the bundle symbolic names of any modules not already listed that you want to prevent Liferay from installing.

   Important: Configuration values must not contain any extra spaces. Extra spaces can short-circuit lists or invalidate the configuration entry.

5. Copy the configuration file into the `[Liferay_Home]/osgi/configs` folder to deploy it.

Note: Blacklisting an LPKG uninstalls all of its internal modules.

Blacklist Bundle Symbolic Names

Type	Bundle Symbolic Name
Module JAR	`Bundle-SymbolicName` in `bnd.bnd` or `MANIFEST.MF` file
LPKG	LPKG file name without the `.lpkg` extension

Type	Bundle Symbolic Name
WAR	Servlet context name in `liferay-plugin-package.properties` file or the WAR file name (minus `.war`), if there is no servlet context name property

Liferay DXP removes installed modules on the blacklist. Blacklisted modules can't be installed. The Liferay DXP log reports each module uninstallation.

Reinstalling Blacklisted Modules

To reinstall and permit installation of blacklisted modules, follow these steps:

1. Open the configuration file `com.liferay.portal.bundle.blacklist.internal.BundleBlacklistConfiguration.config`.
2. Remove the symbolic names of the modules, LPKGs, or WARs from the `blacklistBundleSymbolicNames` list and save the file.

To reinstall *all* the blacklisted modules execute one of these options:

- Remove the configuration file.
- Uninstall the module `com.liferay.portal.bundle.blacklist` using the Application Manager or Felix Gogo Shell.

Note: To temporarily reinstall a module that's been blacklisted, you can remove its symbolic name from the Bundle Blacklist module in System Settings and click the *Update* button. If you want the module to install on subsequent Liferay server startup, make sure to remove the module's symbolic name from any existing module blacklist configuration file in the `[Liferay_Home]/osgi/configs` folder.

The Liferay DXP log reports each module installation.

Blacklisting Components

Follow these steps to blacklist components:

1. In the Control Panel, navigate to *Configuration → System Settings → Component Blacklist Configuration*.
2. Add the names of components to disable and click on the *Update* button.
3. To export the blacklist, click on the Component Blacklist Configuration module's actions button (⋮) and then click *Export*. The blacklist configuration file `com.liferay.portal.component.blacklist.internal.ComponentBlac` downloads. Here are contents from an example file:

```
blacklistComponentNames=["com.liferay.portal.security.ldap.internal.authenticator.LDAPAuth","com.liferay.ip.geocoder.sample.web.internal.portlet.IPG
```

4. Add the names of any components not already listed (e.g., components of modules not yet installed) that you want to prevent Liferay from activating.

   Important: Configuration values must not contain any extra spaces. Extra spaces can short-circuit lists or invalidate the configuration entry.

5. Copy the configuration file into the `[Liferay_Home]/osgi/configs` folder to deploy it.

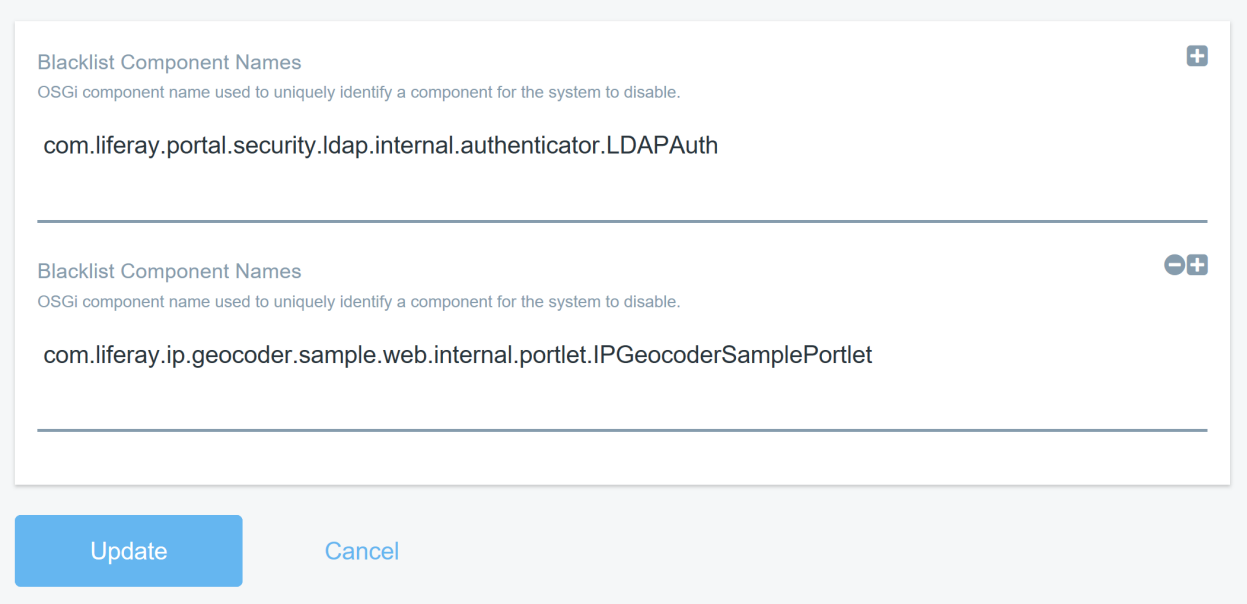


Figure 29.10: This blacklist disables the components `com.liferay.portal.security.ldap.internal.authenticator.LDAPAuth` and `com.liferay.ip.geocoder.sample.web.internal.portlet.IPGeocoderSamplePortlet`.

Re-enabling Blacklisted Components

To re-enable and permit enabling of blacklisted components, follow these steps:

1. Open the configuration file `com.liferay.portal.component.blacklist.internal.ComponentBlacklistConfiguration.conf`
2. Remove the names of the components from the `blacklistComponentNames` list and save the file.

To enable *all* the blacklisted components remove the configuration file.

Note: To temporarily reactivate a component that's been blacklisted, you can remove its name from the Component Blacklist Configuration module in System Settings and click the *Update* button. If you want the component to activate on subsequent Liferay server startup, make sure to remove the component's name from any existing component blacklist configuration file in the `[Liferay_Home]/osgi/configs` folder.

Now you know how to use the blacklisting features for modules and components.

Chapter 30

Configuring Liferay DXP

If Liferay DXP is anything, it's configurable. As the core of Liferay is shrinking due to its increased modularity, it's important that all the applications in Liferay are also configurable.

Breaking it down, three types of applications must be configurable:

1. Liferay itself
2. Liferay's native applications
3. Custom applications

To this end, Liferay's engineers have made the platform and its own applications configurable, and created a mechanism for developers to make their own applications configurable.

In this article you'll learn how to mix the usual (`portal-ext.properties`, anyone?) configuration options with some new ones (*System Settings* in the Control Panel, for example) to configure Liferay, Liferay's applications, and any applications added by your enterprise's developers.

30.1 Where Configuration Happens

Liferay's configuration takes place in the following places:

Properties files: properties files that set default behavior may be included in the platform or the modules. Keep in mind that these settings can always be overwritten by a system administrator in the UI.

Database: configuration through Liferay's UI is stored in a database. The values in the database always override the default configurations set through properties files.

So where in the UI do these configuration options appear? Well, that depends on the scope you want to affect with the settings you choose.

At what level do you want to configure Liferay and its apps? Find out about configuration scope next.

30.2 Configuration Scope

Scope is an important concept to understand when configuring Liferay. Take Language settings, for example. You can set the default language used by the portal instance. You can also set the default language of a site. Some applications even let you set the default language used by a specific piece of content.

Depending on the scope you choose, you'll impact Liferay and its applications with more or less granularity. At one end of the spectrum, you can affect the scope of the whole *System*. Configurations made at the

System scope affect all portal instances, sites, and portlet instances. At the opposite end of the spectrum, configurations made at the *Portlet Instance* level provide configuration settings only for that particular instance of the portlet. For details on configuring a scope for a particular application instance, visit the Application Scope article. Here's an overview of the available configuration scopes:

System: configuring Liferay and its applications through System Settings provides default settings for all portal instances, sites, or portlet instances.

Portal Instance: Configuring Liferay in Instance Settings provides settings that act on the specific portal instance for which they are made, including sites and portlet instances in the portal instance.

Site: Configurations made at the site scope, where you select the site to configure in the site selector, provide settings that take place only in that site. Alternate configurations can be made in different sites.

Portlet Instance: configuring a specific portlet instance only provides a configuration for that particular instance.

Scopes in Liferay are hierarchical so that one scope can set the default values for the underlying subscopes. For example, it is possible to set the default values for all portal instances, sites or portlet instances at configuration at the system level. Think of the settings made at higher levels as being defaults. If a different configuration setting is made at a level with more granularity (for example, the portlet instance), it takes precedence over the settings made at less granular scopes (for example, the portal instance scope).

This section contains articles on configuring Liferay at the System and Instance scopes:

System wide configuration:

- System Settings
- Server Administration
- Custom Fields

Setting up a portal instance:

- Virtual Instances
- Instance Settings
- Integrating Existing Users

All of these are accomplished through the Control Panel. Start by learning to configure modules system-wide in the System Settings Control Panel app.

Chapter 31

System Wide Settings

Liferay is so configurable it can be hard to keep track of where all the configuration interfaces reside. Fortunately, the Control Panel is most often the place to look, and there's a tidy *Configruation* section there that houses a lot of the higher level (for example, system and instance scoped) configuration options. This section considers the configuration options dealing with the *System* scope. What is the system scope? It's the entire system, of course. That might sound self explanatory, and it is, for the most part. It just means that configuration at this level affects the entire system, which means all the *Virtual Instances* of Liferay in the system. If you're not sure what a Virtual Instance is, read the section on Setting Up a Liferay Instance.

There are several configuration sections that take effect at the system scope:

- System Settings
- Server Administration
- Custom Fields

Get started by learning about System Settings.

31.1 System Settings

You might be tired of hearing about it by now, but it's worth repeating: Liferay DXP is modular. It's composed of many applications, which are divided into even smaller "chunks" of functionality. The applications, and sometimes even code chunks, are configurable at several scopes, as discussed in the introductory article for this section.

If you need to make configuration changes to an application that take effect in a system-wide fashion, where do you do it? Readers who have been around a while will be raising their hands hoping to be called on, sure that the correct answer is "why, system-wide application configuration takes place in a properties file, most commonly the `portal-ext.properties` file". They then expect to get a sticker of Ray on their mobile device for their attentiveness. However, that's an incomplete answer, so they need to sit back down and listen like the rest of the class. There's a brand new way to make configuration changes at the system scope in 7.0, and you don't need to go messing around in a properties file to do it. These settings can now be made in the Control Panel, in *Configuration → System Settings*.

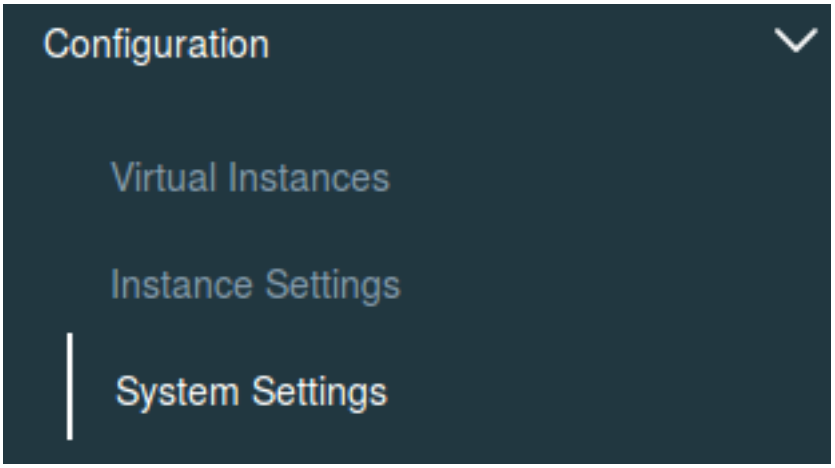


Figure 31.1: System Settings are accessed through the Control Panel.

Editing the Default Configuration

In System Settings, the configuration options are categorized into logical groupings based on their functionality, and there's a Search box to make finding the app you want to configure easier.

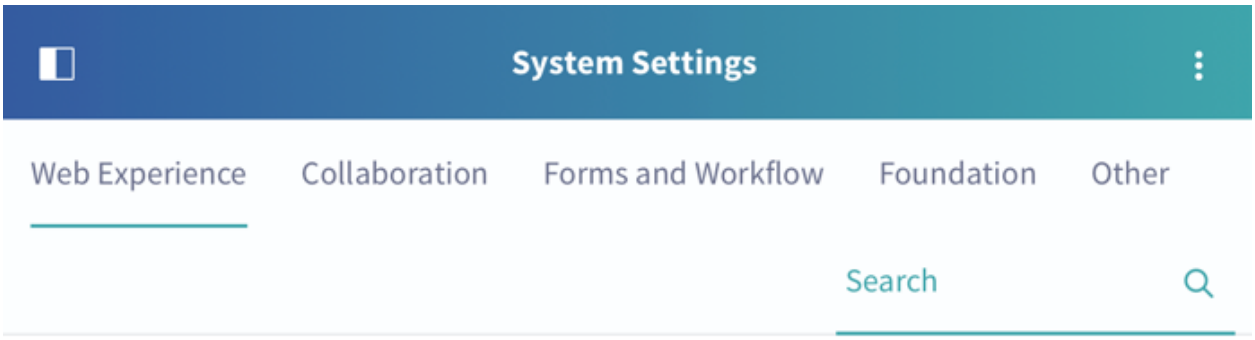


Figure 31.2: System Settings are organized by component.

Changing an app's default configuration options in System Settings isn't difficult. Once you find what you're looking for, simply click the name of the component you want to configure, or click the actions button (⋮) then click *Edit*. Make any changes you'd like, then click *Save*. Your configuration changes are saved and applied throughout the system.

Important: Content generated using templates (e.g., FreeMarker templates and Application Display Templates) is cached. Cached content might not reflect configuration changes until the cache is invalidated (cleared). The Server Administration → Resources tab provides cache clearing options.

If you make some regrettable configuration decisions and can't recall exactly what you did, start over by clicking the actions button (⋮), then clicking *Reset Default Values*.

Figure 31.3: After saving changes to a configuration, the actions *Reset Default Values* and *Export* are available for it.

Configuration Scope

While browsing the categories of System Settings, you'll notice that each entry has a *Name* and a *Scope*. This scope is different than the overall application configuration scope discussed in the introduction to this section. This is the System Settings application after all, so all of the configuration done here affects the System scope. The Scope field here refers to the scope at which this default, system-wide configuration can be overridden. There are four values that you'll see under Scope:

- *System:* Any configuration made at the system scope here becomes the final value for the application in a system-wide fashion. It affects the whole system and cannot be overridden anywhere else.

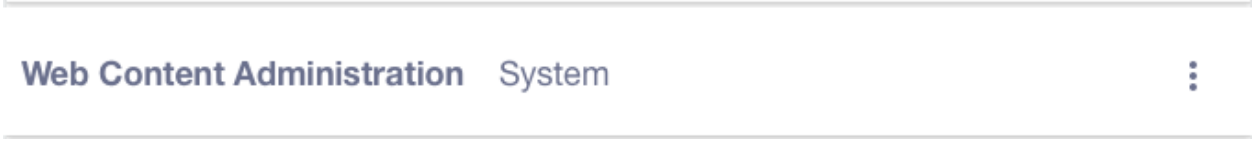


Figure 31.4: Some System Settings entries have a System scope.

- *Default Configuration for Application:* Making configuration changes at this level affects the application in a system-wide fashion as well. However, while these become the system-wide defaults for the application, they can be overridden from each application instance.

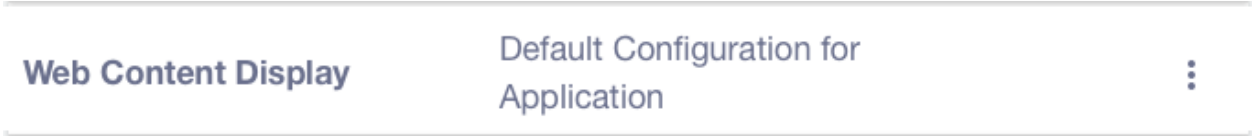


Figure 31.5: Some System Settings entries have a Default Configuration for Application scope.

- *Default Configuration for All Sites:* Configuration at this scope can be overridden in each site.
- *Default Settings for All Instances:* Configuration at this scope can be overridden in each Liferay DXP instance (usually via Instance Settings).

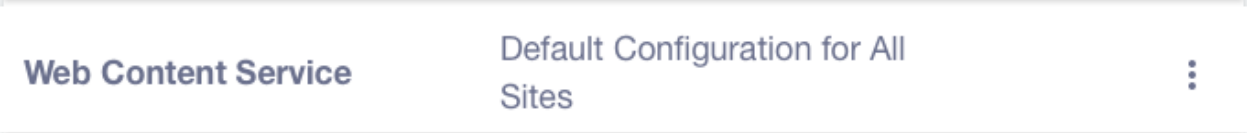


Figure 31.6: Some System Settings entries have a Default Configuration for All Sites scope.

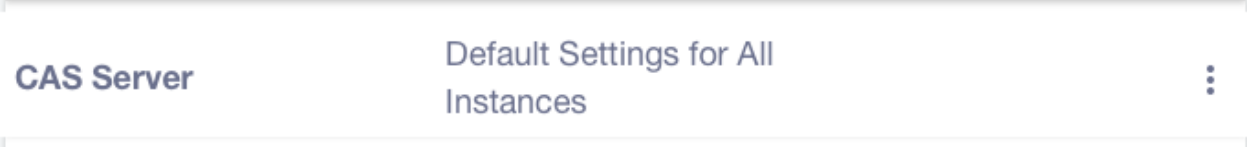


Figure 31.7: Some System Settings entries have a Default Configuration for All Instances scope.

If an application is configurable from Instance Settings and System Settings, use System Settings to configure it whenever possible. If you use Instance Settings and later want to revert to using the System Settings default configurations, look for the *Reset Values* button from the Instance Settings entry.

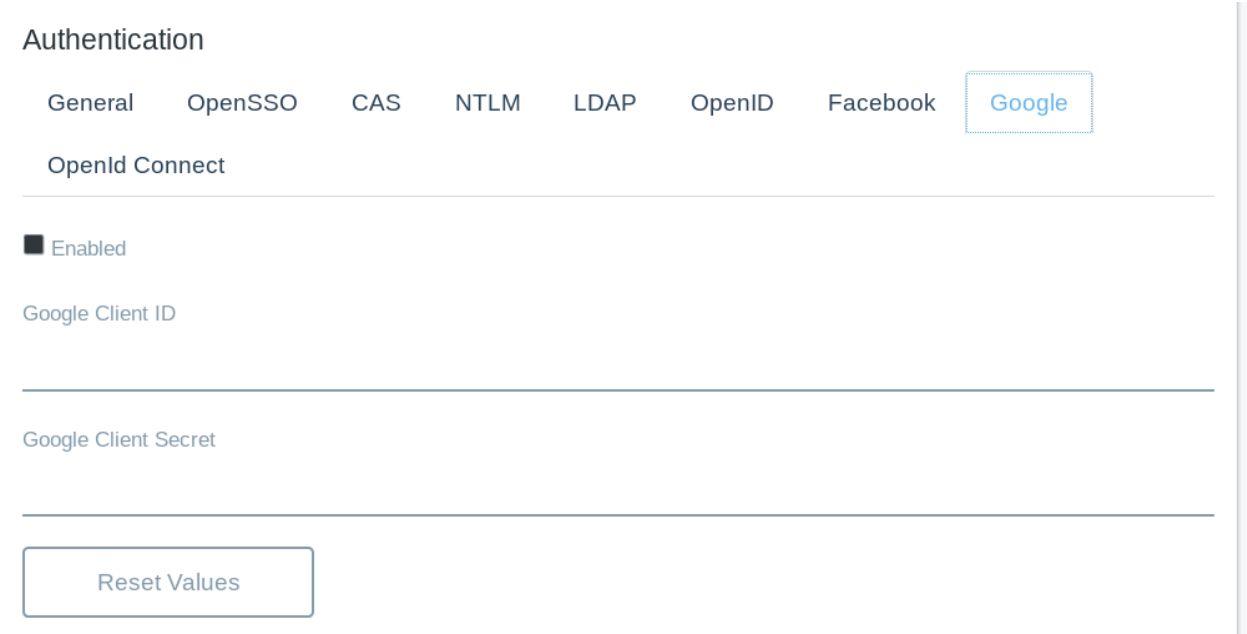


Figure 31.8: Some Instance Settings entries have a *Reset Values* button so you can safely revert your configuration changes, falling back to the System Settings defaults.

The Reset Values button removes configuration values from the Liferay DXP database so you can rely on the defaults specified in System Settings. If there is no Reset Values button for an Instance Settings entry, once you use Instance Settings to configure the application, you can't later decide to use System Settings. Only changes to Instance Settings will be recognized.

Exporting and Importing Configurations

What if you change many default configurations in System Settings, and then need to make the same changes in another Liferay DXP installation? Don't worry, you don't need to remember and duplicate every choice you made. The System Settings application lets you export a single entry's configurations (e.g., the Dynamic Data Lists entry's configurations), or all the settings you made in the System Settings interface. Then you can deploy them in the target Liferay DXP installation.

To export a single entry's configurations, click the actions button (⋮), then click *Export*. A `.config` file then downloads that contains any configuration changes you saved.

Note: If you're running Liferay DE 7.0 Fix Pack 7 or earlier, or Liferay CE Portal 7.0 GA3 or earlier, System Settings exports a `.cfg` file instead of a `.config` file. The file format was changed to fix a bug with multi-value settings, which the `.cfg` format didn't handle properly. See the bug report for more information. Any Liferay DXP system newer than those listed in this note uses the `.config` file format.

Although `.config` files are now the default file type exported by System Settings, you're free to continue using `.cfg` files–they're still supported by the configuration framework underlying System Settings.

To export all the configuration changes you've made in System Settings, click the System Settings options button (⋮), then click *Export All Settings*. The `.config` files for all the entries you edited then download in a ZIP file.

To make these configurations active in the destination Liferay DXP system, simply unzip and place the `.config` files in the `[Liferay_Home]/osgi/configs` folder.

Awesome! Now you know what System Settings is and how to use it. All that's left is to explore the entries to see what configuration options you can make. If you aren't sure what something does, check the documentation for the feature you're interested in, as specific configurations are covered there.

31.2 Understanding System Configuration Files

Liferay DXP's System Settings application is convenient for making system-scoped configuration changes and setting default configurations for other scopes. But there's another supported configuration approach: configuration files. You can use configuration files to transfer configurations from pre-production systems to production systems, or between any other Liferay DXP systems. Sometimes developers choose to distribute the default configuration for their applications via configuration file. Whatever the reason, configuration files are a straightforward way of configuring Liferay DXP.

Creating Configuration Files

System Settings provides an *Export* option that becomes available once you modify a configuration entry. Exporting is the recommended way to create `.config` files: you download a `.config` file containing the entry's settings in a `key=value` format. Liferay DXP exports an entry's total available configuration keys and values, even if only one value was changed. You can export a single configuration entry or the entire set of modified configurations in Liferay DXP.

To avoid a file name conflict, name configuration files using a unique identifier. For example, the Journal Service entry, which backs Liferay's Web Content capabilities, has this file name:

```
com.liferay.journal.configuration.JournalServiceConfiguration.config
```

Configuration files use the `.config` property value format defined by the Apache Felix Configuration Admin framework.

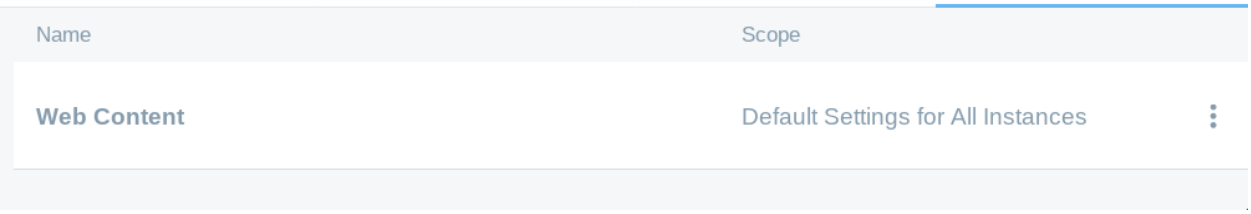


Figure 31.9: The Web Content System Settings entry has the backend ID `com.liferay.journal.configuration.JournalServiceConfiguration`.

Important: Content generated using templates (e.g., FreeMarker templates and Application Display Templates) is cached. Cached content might not reflect configuration changes until the cache is invalidated (cleared). The Server Administration → Resources tab provides cache clearing options.

The `.cfg` file format is common in OSGi environments, so Liferay DXP supports it, but `.config` files are preferable. You can specify a property value's type, and have multi-valued properties. The syntax described below is for `.config` files.

Key/Value Syntax The general syntax for all keys and values in a `.config` file is the same:

```
configurationName="value"
```

For single value configurations without special characters, that's all there is to know. Settings with multiple values and certain characters require slight modifications.

Multi-Value Settings Configuration entries can have properties that accept multiple values. For example, a configuration property for specifying supported file extensions needs more than one value. Here's how to write a multi-value setting in a `.config` file:

```
multiValueSetting=["Value 1","Value 2", ...]
```

Do not use a space character between values (after the comma). An errant space character can cause a failure to load the property.

Open the Web Content entry from System Settings and you'll see what looks like multiple single value entries for *Charactersblacklist*:

In the configuration file, this is really a single key with an array of comma-separated values:

```
charactersblacklist=["&","'","@","\\","]","}",":","\=",">","/","<","[","{","%","+","#","`","?","\"",";","*","~"]
```

Escaping Characters Double quotes (") and equals signs (=) must be *escaped* in `.config` files. Escaping is using another character to denote that a character shouldn't be used in its normal way. Since double quotes and equals signs are already used in `.config` files, escaping them tells the framework not to read them the normal way, but to pass them through as part of the value. Use a \ to escape characters in the `.config` file:

```
charactersblacklist=["&","\"","\="]
```

This setting illustrates a multi-value setting with a regular, unescaped character (&), and two escaped ones (\" and \=).

Along with the mandatory escaping of double quotes and equals characters, it's beneficial to escape spaces inside values to avoid problems.

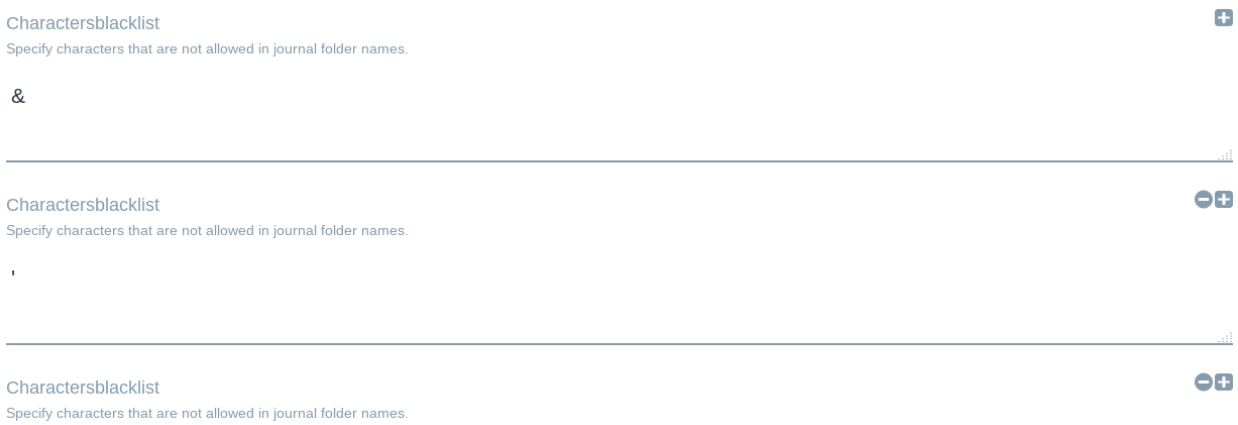


Figure 31.10: The Web Content System Settings entry has many *Charactersblacklist* fields.

```
blacklistBundleSymbolicNames=["Liferay\ Marketplace","Liferay\ Sharepoint\ Connector"]
```

In the above example, a \ is used before each space character to ensure it's read and processed properly. If you don't escape spaces yourself, the framework adds the backslash for you after deployment.

Deploying a Configuration File Once you have a configuration file, deploy it so Liferay DXP recognizes it and updates the targeted configuration values.

To deploy the `.config` file, place it in your Liferay Home's `osgi/configs` folder. To change the configuration further, you can edit the `.config` file directly or use System Settings.

Typed Values The `.config` file format supports specifying the type of a configuration value by inserting a special type marker character. Because Liferay DXP already knows the correct type for each configuration property, the type characters are only useful for informational purposes. For example, a configuration with a boolean type has *B* just before the value to mark it as a boolean type:

```
addDefaultStructures=B"true"
```

If you see type markers in `.config` files, you can safely ignore them. The example included above functions identically without the type marker:

```
addDefaultStructures="true"
```

Configuration Files and Clustering

In a clustered environment, each node needs the same configuration values for each entry. For example, all nodes should use the same *Blogs* configuration settings. To accomplish this, deploy a `.config` file to *one* node. Liferay DXP uses an internal system that appies the change to all nodes in the cluster.

Factory Configurations

Configurations supporting multiple independent occurrences of the same configuration entry are called *factory configurations*.

Factory Configuration Example: Liferay DXP supports the publication of JAX-WS and JAX-RS web services. These services must use a CXF Endpoint, which is a context path where the web services are deployed and accessed. Endpoints can be created via factory configuration by navigating to the CXF Endpoints System Settings Entry (System Settings → Foundation → CXF Endpoints). Using the Add button ( ), enter the desired configuration values and repeat the process, adding as many CXF Endpoint configurations as needed. Creating CXF Endpoint configurations also creates CXF Endpoints themselves. This is how factory configuration works.

If a service is meant to support factory configuration, its System Settings entry has an Add button ( ).

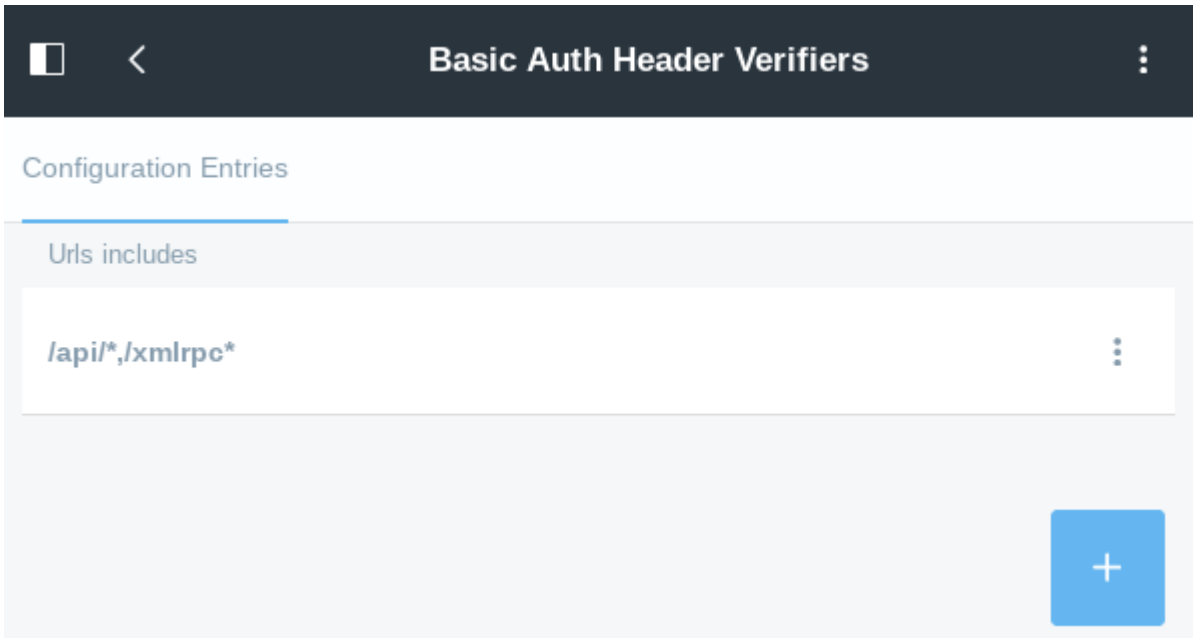


Figure 31.11: If a System Settings entry has an Add button, it's suitable for factory configuration.

As with single-instance configurations, you can accomplish factory configuration in the System Settings interface (as described in the example above) or via configuration files. Once you determine you must write a factory configuration file. Name a standard single-instance configuration file like this:

```
my.service.ServiceConfiguration.config
```

If your service supports factory configuration, use Liferay's convention of calling the first instance of the configuration `-default.config`:

```
my.service.ServiceConfiguration-default.config
```

The next instance contains a unique *subname* (something other than *default*). It's smart to use a descriptive name:

```
my.service.ServiceConfiguration-port9080.config
```

To follow the CXF Endpoints example described above, if Liferay's developers had shipped an initial CXF Endpoint `.config` file with Liferay DXP, it would have been named

```
com.liferay.portal.remote.cxf.common.configuration.CXFEndpointPublisherConfiguration-default.config
```

Perhaps the `-default.config` configuration specifies a context path for *REST* web services, and then you create another endpoint with a different context path for *SOAP* web services. Your second configuration file could be named

```
com.liferay.portal.remote.cxf.common.configuration.CXFEndpointPublisherConfiguration-soap.config
```

Note: Some System Settings entries (like the CXF Endpoints entry) don't ship with a configuration file, so anything you create is the first occurrence. However, if you configure one and export it to obtain the `.config` file, it's not named `-default.config`. Instead it's given a guaranteed unique identifier for its subname, like

```
com.liferay.portal.remote.cxf.common.configuration.CXFEndpointPublisherConfiguration-a6f67e48-6dca-49c6-bf6b-8fd5e6016b2d.config
```

This is to guarantee that the file has a unique name. If you're exporting the configuration file for deployment in a *separate Liferay DXP system*, you can rename the exported file to use a more descriptive subname. If you rename the file and deploy it to the same system it was exported from, the new subname marks it as an entirely new configuration. You'll end up with an additional configuration instance in this case, not just a renamed one.

Warning:: For configuration entries supporting factory configuration, omitting the sub-name from a `.config` file's name breaks the System Settings application's functionality (but only for the configuration entry targeted by the `.config` file). This is caused by a known bug. See LPS-76352 for more information. Once an improperly named configuration file is deployed, you can't add any entries for the configuration in question from its System Settings entry. For example, if you deploy a

```
com.liferay.portal.remote.cxf.common.configuration.CXFEndpointPublisherConfiguration.config
```

file to configure a CXF Endpoint, not only does this not add a CXF Endpoint, it also prevents you from adding any CXF Endpoints via System Settings.

Deploying an erroneous (lacking a sub-name) `.config` file doesn't break anything permanently. Just rename the file using the proper convention described above or remove it entirely and start over.

Now a note of warning. In many cases, configuration files can be used to force a factory configuration scenario, but not all configurations are designed to be used this way. It's best to stick to the intended use cases. Use System Settings as described above to determine if factory configuration is a good idea. If not, stick to the single occurrence mode of configuration (specifying only one configuration file for the service).

31.3 Environment Variables

You can use environment variables to configure Liferay DXP, which is especially useful in cloud-based environments.

Important: This capability is for Linux and Unix operating systems only: it's not available on Windows.

Note: Environment variables can only override portal properties (defined in `portal.properties`); they cannot override system properties or configurations from `.config` files.

Overriding Portal Properties

To override portal properties, follow these operating system (OS) and Liferay environment variable name guidelines:

OS environment variable names: - Consist of uppercase letters, digits, and _ (underscore) characters only - Must not start with a digit

Liferay environment variable names: - Must follow the OS environment variable name guidelines - Must start with the prefix `LIFERAY_` - Characters that violate the OS guidelines must be encoded using one of the following methods: - `CharPool` constants - Unicode code points

Encoding with CharPool Constants Follow these rules to encode with `CharPool` constants:

- Use upper case (`CharPool` constant names are uppercase).
- Remove _ characters from the `CharPool` constant names.
- Wrap each constant name in a pair of _ characters.

Here are some encoding examples.

CharPool Constant Example 1 This example involves encoding the period `.` character.

Portal property:

```
setup.wizard.enabled=false
```

Encoding steps:

1. Prefix the variable with `LIFERAY_`.

   `LIFERAY_setup.wizard.enabled=false`

2. Encode the period characters with the `CharPool` constant name `PERIOD`, surrounded by a pair of _ characters.

   `LIFERAY_setup_PERIOD_wizard_PERIOD_enabled=false`

3. Convert all letters to uppercase.

Environment variable:

```
LIFERAY_SETUP_PERIOD_WIZARD_PERIOD_ENABLED=false
```

CharPool Constant Example 2 This example uses `CharPool` constants to encode an environment variable name for a property that includes periods, brackets, and uppercase characters.

Portal property:

```
setup.database.driverClassName[mysql]=com.mysql.jdbc.Driver
```

Encoding steps: 1. Prefix the variable with `LIFERAY_`.

```
`LIFERAY_setup.database.driverClassName[mysql]=com.mysql.jdbc.Driver`
```

2. Use `CharPool` constant names surrounded by _ characters:

   - `.` → `_PERIOD_`

- C → _UPPER_CASE_C_
- N → _UPPER_CASE_N_
- [→ _OPEN_BRACKET_
-] → _CLOSE_BRACKET_

```
LIFERAY_setup_PERIOD_database_PERIOD_driver_UPPER_CASE_C_lass_UPPER_CASE_N_ame_OPEN_BRACKET_mysql_CLOSE_BRAC
```

3. Remove underscores from all constant names.

   - _UPPER_CASE_C_ → _UPPERCASEC_
   - _UPPER_CASE_N_ → _UPPERCASEN_
   - _OPEN_BRACKET_ → _OPENBRACKET_
   - _CLOSE_BRACKET_ → _CLOSEBRACKET_

   ```
   LIFERAY_setup_PERIOD_database_PERIOD_driver_UPPERCASEC_lass_UPPERCASEN_ame_OPENBRACKET_mysql_CLOSEBRACKET_=c
   ```

4. Convert all letters to uppercase.

   Environment variable:

```
LIFERAY_SETUP_PERIOD_DATABASE_PERIOD_DRIVER_UPPERCASEC_LASS_UPPERCASEN_AME_OPENBRACKET_MYSQL_CLOSEBRACKET_=com.mysql.jdbc.Driver
```

Encoding with Unicode Code Points Unicode offers more character encodings than `CharPool` and produces shorter variable names. But Unicode code points are cryptic.

Follow these rules to encode with Unicode:

1. Look up the character code points.
2. Replace each character with a code point wrapped in a pair of _ characters.

Here's an encoding example that uses Unicode.

Unicode Example This example uses code points to encode an environment variable name for a portal property that contains periods, brackets, and uppercase characters.

Portal property:

```
setup.database.driverClassName[mysql]=com.mysql.jdbc.Driver
```

Encoding steps:

1. Prefix the variable with `LIFERAY_`.

   ```
   LIFERAY_setup.database.driverClassName[mysql]=com.mysql.jdbc.Driver
   ```

2. Use `CharPool` constant names surrounded by _ characters to encode the following characters:

   - . → _46_
   - C → _67_
   - N → _78_
   - [→ _91_
   -] → _93_

```
LIFERAY_setup_46_database_46_driver_67_lass_78_ame_91_mysql_93_=com.mysql.jdbc.Driver
```

3. Convert all letters to uppercase.

Environment variable:

```
LIFERAY_SETUP_46_DATABASE_46_DRIVER_67_LASS_78_AME_91_MYSQL_93_=com.mysql.jdbc.Driver
```

Congratulations! You know how to use environment variables to configure Liferay DXP. It's easier than ever to configure and deploy Liferay DXP images in virtual machines and cloud containers.

Related Topics

Configuring Liferay DXP
System Wide Settings

31.4 Server Administration

The Server Administration application (accessed by clicking *Control Panel → Configuration → Server Administration*) lets you perform tasks related to the Liferay DXP server. Clicking the link displays a graph showing the resources available in the JVM.

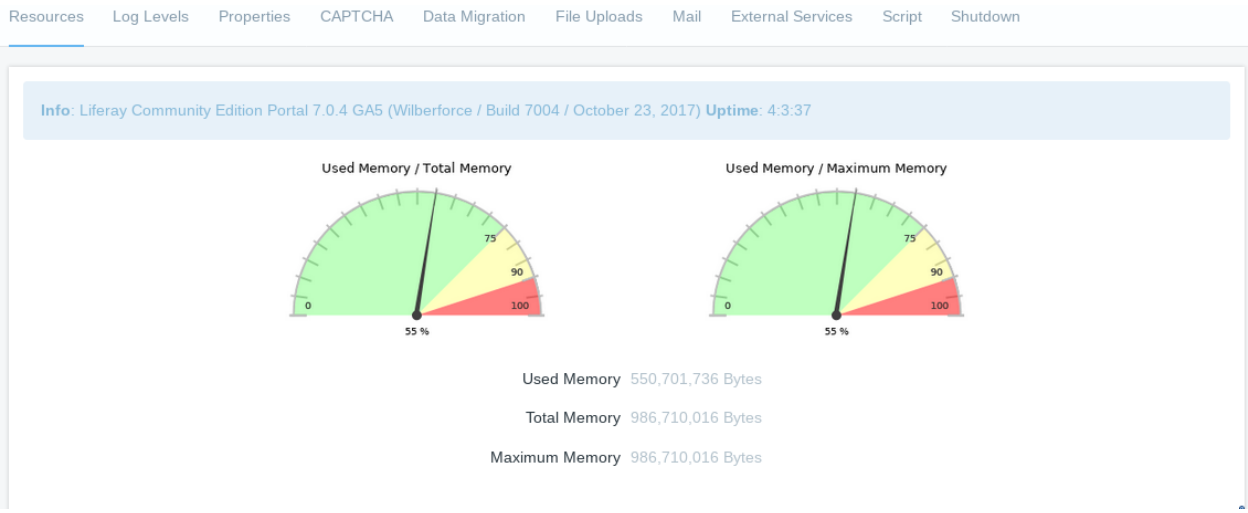


Figure 31.12: Server Administration is where you configure and monitor many aspects of the server.

Note: Liferay DXP customers can use Liferay Connected Services (LCS) to obtain additional performance metrics. If you're a DXP customer, see the LCS documentation for more information.

The Server Administration functionality is broken into several tabs:

- Resources
- Log Levels
- Properties
- CAPTCHA

- Data Migration
- File Uploads
- Mail
- External Services
- Script
- Shutdown

Resources

The Resources tab contains several server wide actions that an administrator can execute. These include the following items:

Run the garbage collector Tells the JVM to free memory by running the garbage collector.

Generate a thread dump Generates a thread dump for later scrutiny to determine the presence and locations of any deadlocks. Useful during performance testing, but you must add a logger category for `com.liferay.server.admin.web.internal.portlet.action.EditServerMVCActionCommand` and set it to `INFO` before executing.

Clear content cached by this VM Clears content stored in the local cache. Only local JVM scope Ehcache content is cleared, not clustered Ehcache. 1

Clear content cached across the cluster Clears the content of the entire clustered Ehcache. 1

Clear the database cache Clears the database cache. Does not clear any Ehcache content except database results at the persistence layer. 1

Clear the direct servlet cache Clears the direct servlet cache. In case emergency fixes must be applied, this action allows an administrator to manually clear out the cache to force JSPs to reload.

The direct servlet context optimizes JSP serving performance by caching and accessing the generated servlets directly instead of accessing them over the application server's dispatcher chain. This function is only suitable for cases where no filter is required for the JSPs; it should be enabled for production mode to improve performance, but disabled for development mode to allow JSP servlets to be reloaded on the fly. See the Direct Servlet Context section of the `portal.properties` file for details. 1

Reindex all search indexes Regenerates all search indexes. If you are not using a Solr search server this impacts portal performance so try to do this at non-peak times.

Reindex all spell check indexes Regenerates all spell check indexes.

Reindex com.liferay...SomeEntity Reindexes a single entity.

Verify database tables of all plugins Checks all tables against their indexes for data retrieval accuracy.

Verify Membership Policies Checks that existing site membership policies were correctly applied and automatically makes updates. If the Liferay database is changed manually or is hacked–resulting in a user assigned to a site in violation of a site membership policy–this action triggers the verification methods of all implemented site membership policies. Changes are automatically made to any memberships in violation.

Reset preview and thumbnail files for the Documents and Media portlet Regenerates previews of each item in your Documents and Media libraries.

Clean up permissions Removes permissions on the Guest, User, and Power User roles to simplify the management of "User Customizable Pages". The "Add To Page" permissions is removed from the Guest and User roles for all portlets, and is reduced in scope for Power Users from portal-wide to "User Personal Site."

Clean up portlet preferences This action cleans up database entries if portlet preferences become orphaned in the Liferay DXP database.

[1] Caching occurs at multiple levels. Some higher caching layers aren't aware of lower caching layers. Always clear the cache at the lowest (most granular) layer possible, even if you've already cleared a higher level cache.

Log Levels

Here, you can make dynamic modifications of log levels for any class hierarchy in Liferay DXP. Custom objects not on the list can be added with the *Add Category* tab. Changes to the log level near the top of the class hierarchy (such as at `com.liferay`) also changes log levels for all the classes under that hierarchy. Modifications unnecessarily high in the hierarchy generate too many messages to be useful.

Properties

Liferay and the JVM contain many settings which are defined as properties. The properties tab has two sub-tabs: System Properties and Portal Properties.

The system properties tab shows an exhaustive list of system properties for the JVM, as well as many Liferay DXP system properties. This information can be used for debugging purposes or to check the currently running configuration.

The portal properties tab shows an exhaustive list of the current portal property values, so you don't have to shut down Liferay DXP or open the properties file directly. Portal properties are customizable; you can peruse the full list of available portal properties.

CAPTCHA

CAPTCHA is designed to thwart bots from registering for accounts. By default, Liferay DXP ships with its own CAPTCHA service.

Alternatively, Google's reCAPTCHA service can be enabled from the CAPTCHA tab of Server Administration. Check the *Enable ReCAPTCHA* box, enter your public and private keys into the provided fields, and click *Save*. Liferay DXP then uses reCAPTCHA instead of the native CAPTCHA solution.

Data Migration

This tab is for upgrading Liferay DXP or migrating its data from one system to another without writing custom scripts.

The first section, under *Migrate data from one database to another.* was useful in past versions of Liferay DXP for copying the entire Liferay DXP database to a new database. However, the database migration tool cannot be used in the current version of Liferay DXP and will be removed in the next version of Liferay DXP.

The second section, *Migrate documents from one repository to another*, helps you migrate your documents to a new repository on a different disk or in a new format. Here are the steps:

1. Create a backup copy of the Document Library repository and Liferay DXP database.
2. Configure the new file store on that file store's page, listed under *System Settings → Foundation*.
3. In this tab (*Server Administration → Data Migration*), select the repository hook for the file store you configured and click *Execute*.
4. After the process completes, configure the new repository as the default.
5. If you used a `portal-ext.properties` file to configure the repository, restart the server.

File Uploads

Since Liferay DXP allows users to upload files in various places, you may want to restrict the type and size of files users are allowed to upload. The File Uploads tab lets you set the overall maximum file size and then grant exceptions to that limit for specific applications. You can also limit the allowed file extensions generally or by application.

Mail

Instead of using a `portal-ext.properties` file to configure a mail server, you can configure a mail server from the Mail tab. If your message boards receive mail, you can connect a POP mail server. If Liferay DXP sends mail (useful for sending notifications to users), you can connect to an SMTP server. We highly recommend setting up mail servers.

Note that if you configure mail server settings here in the Control Panel, these settings override any mail server settings in your `portal-ext.properties` file.

External Services

Liferay DXP lets users upload and share content via the Documents and Media library, a customizable and permissions-enabled online repository for files. Users can upload files of any type to the Documents and Media library. Liferay DXP ships with PDFBox for generating automatic previews for certain types of documents. You can install three additional tools that offer higher quality previews and document conversion functionality: OpenOffice or LibreOffice, ImageMagick and Xuggler. With Liferay DXP configured to use these tools, you can generate automatic previews for many types of files including text files, office suite files, PDFs, images, audio files and videos. Users can also use the conversion functionality to download documents in a variety of formats. Please see the article on publishing files in Liferay DXP.

LibreOffice is available here: LibreOffice, ImageMagick is available here: ImageMagick, and Xuggler is available here: Xuggler. Make sure to choose the correct versions of these applications for your operating system. We recommend that you install the latest stable versions. LibreOffice and ImageMagick must be installed manually, but you can install Xuggler from the Control Panel.

Tip: If you're running Liferay DXP on a Linux server and experience a problem enabling Xuggler, check your server's glibc version. You might have to update glibc to version 2.6 or later in order for Xuggler to work.

Once you've installed these tools, you can use the Control Panel's External Services tab to configure Liferay DXP to use them.

OpenOffice/LibreOffice Configuration OpenOffice and LibreOffice are open source office suites which are usually run in graphical mode to create documents, but they can also be run in server mode. When run in server mode, OpenOffice and LibreOffice can be used to convert documents to and from all of the file types it supports. Once configured, Liferay DXP uses this feature to convert content on the fly automatically. You can install OpenOffice or LibreOffice on the same machine where Liferay DXP is running or you can connect to a separate host.

Note: Running OpenOffice or LibreOffice in server mode on a remote host is not recommended for converting documents in Liferay DXP, and is not fully supported. To avoid unexpected behavior during document conversions, you should run OpenOffice or LibreOffice in server mode on the same machine as Liferay DXP.

You can start LibreOffice or OpenOffice in server mode with the following command:

```
soffice --headless --accept="socket,host=127.0.0.1,port=8100;urp;"
--nofirststartwizard
```

Once OpenOffice or LibreOffice is installed and running in server mode, configure Liferay DXP to use it either in `portal-ext.properties` or from the Control Panel. To enable OpenOffice/LibreOffice in your `portal-ext.properties` file, add the following line:

```
openoffice.server.enabled=true
```

If OpenOffice or LibreOffice is running on another server or on a non-default port, you must also specify these values. The default values are as follows:

```
openoffice.server.host=127.0.0.1
openoffice.server.port=8100
```

By default, when Liferay DXP uses OpenOffice or LibreOffice to perform conversions, it uses a cache. The first time a document is converted, a copy is saved in the Liferay DXP temp folder `/liferay/document_conversion/`. When Liferay DXP receives a conversion request, it checks this folder to see if the converted document already exists. If the converted document is found, Liferay DXP returns it to the user. Otherwise, it performs a fresh conversion and saves a copy in the temp folder. If the cache is turned off, Liferay DXP always regenerates the file even if a previously existing conversion already exists in the temp folder. You can turn the cache off by setting the following property:

```
openoffice.cache.enabled=false
```

To configure Liferay DXP to use OpenOffice/LibreOffice from the Control Panel, navigate to the *Server Administration → External Services* page and check the *Enabled* box for OpenOffice. If OpenOffice/LibreOffice is running on a non-default port–the default is port 8100–you must also specify the port number. If you have something else running on this port, find an open port and specify it both in the command to start OpenOffice/LibreOffice in server mode and on the Control Panel's External Services configuration page. When you are finished, click *Save*. Now Liferay DXP can perform many types of document conversions.

ImageMagick configuration Before configuring ImageMagick to generate image and PDF previews, install it and its dependency, Ghostscript. This differs by operating system: on Linux, both are likely already installed. They are not likely to be installed on Windows but may be on Macs.

1. Download and install *ImageMagick*.

2. Download and install *Ghostscript*.

Once installed, enable ImageMagick through the Control Panel or a `portal-ext.properties` file. If using `portal-ext.properties`, add the following lines and make sure the search path points to the directories containing the ImageMagick and Ghostscript executables. You may also need to configure the path for fonts used by Ghostscript when in Mac or Unix environments.

```
imagemagick.enabled=true
imagemagick.global.search.path[apple]=/opt/local/bin:/opt/local/share/ghostscript/fonts:/opt/local/share/fonts/urw-fonts
imagemagick.global.search.path[unix]=/usr/local/bin:/usr/local/share/ghostscript/fonts:/usr/local/share/fonts/urw-fonts
imagemagick.global.search.path[windows]=C:\\Program Files\\ImageMagick
```

To enable ImageMagick from the Control Panel:

1. Navigate to the *Server Administration → External Services* page.
2. Check *Enabled* for ImageMagick and verify the paths to the ImageMagick and Ghostscript executables are correct.

Some older versions of ImageMagick don't run properly with Liferay DXP. If this is the case, update to the latest version (ImageMagick 6.7.9-6 2012-09-25 Q16 or later). To check for the latest ImageMagick versions, visit ImageMagick's website. See LPS-30291 for information on efforts to identify incompatible application versions with Liferay DXP.

Xuggler configuration From the control panel, navigate to the *Server Administration → External Services* page. From the dropdown, select the Xuggler `.jar` file that matches your operating system. Then click on *Install*.

After installing, restart your application server and enable Xuggler for your portal. You can enable Xuggler either from the Control Panel or by specifying the portal property that enables Xuggler.

To enable Xuggler from portal properties, add the following line to your `portal-ext.properties` file and restart your application server.

```
xuggler.enabled=true
```

To enable Xuggler from the Control Panel, navigate to the *Server Administration → External Services* page, check *Enabled*, and click on *Save*.

That's it! You've successfully configured Documents and Media to use Xuggler for audio and video files.

Script

The Script tab includes a scripting console for executing migration or management code. The Groovy scripting language is supported out of the box. For more information, see the Scripting article for more information on how to use the scripting console and examples.

Shutdown

If you ever need to shut down your Liferay DXP server while users are logged in, you can use the Shutdown tab to inform your logged-in users of the impending shutdown. You can define the number of minutes until the shutdown and a custom message to display.

Liferay DXP displays the message at the top of users' pages for the duration of time you specified. When the time expires, all pages display a message saying the portal has been shut down. At this point, the server must be restarted to restore access.

Next, let's examine how to manage multiple portal instances.

Chapter 32

Setting Up a Liferay DXP Instance

Once you have a Liferay DXP installation, you're going to want to configure it. Whether that means adding a virtual instance, setting up the initial instance to suit your needs, or integrating existing users into the installation. Recall that configuration happens at different scopes. Here were thinking about the instance scope. There's an important difference between the system scope and the instance scope. The system scope is the highest level scope you can make configurations at. All virtual instances of Liferay are impacted by configuration done at this scope. The instance scope applies only to one particular virtual instance of Liferay. What are virtual instances? Separate Liferay installations (with their own virtual host) that share a server and database. Each instance can be configured very differently.

So what kind of configuration do you need to do in your instance? Here's what these articles cover:

- Adding a virtual instance
- Configuring a virtual instance
- Integrating existing users into Liferay

Get started by learning about adding a virtual instance.

32.1 Virtual Instances

Here's a quick scenario: you already have a server hosting a site created with Liferay. That means you already have a hosted Liferay database. Let's say you need to set up another site that runs on Liferay. While there's more than one way to skin this cat, if you want to kill two birds with one stone, just create an additional Virtual Instance of Liferay. You'll get to have two separate sites (the birds), with a shared server and database (the stone). If you don't mind that the sites have a shared database and server, you can set up a separate Virtual Instance in two shakes of a lambs tail.

It's important to note before proceeding that no real animals were harmed in the writing of the above paragraph (virtual animals, however...).

Now without the idioms: Liferay allows you to run more than one virtual instance on a single server. The Virtual Instances section of the Control Panel appears under the Configuration heading. This section lets you manage multiple Liferay instances from a single Liferay installation. Each instance's data is kept separate from every other instance's data. All instance data, however, is kept in the same database.

Each virtual instance has its own domain name. Liferay directs users to the proper instance based on this domain name. So before you configure an instance, configure its domain name in your network first. When you're ready to add an instance, click the *Add* ( ) button here.

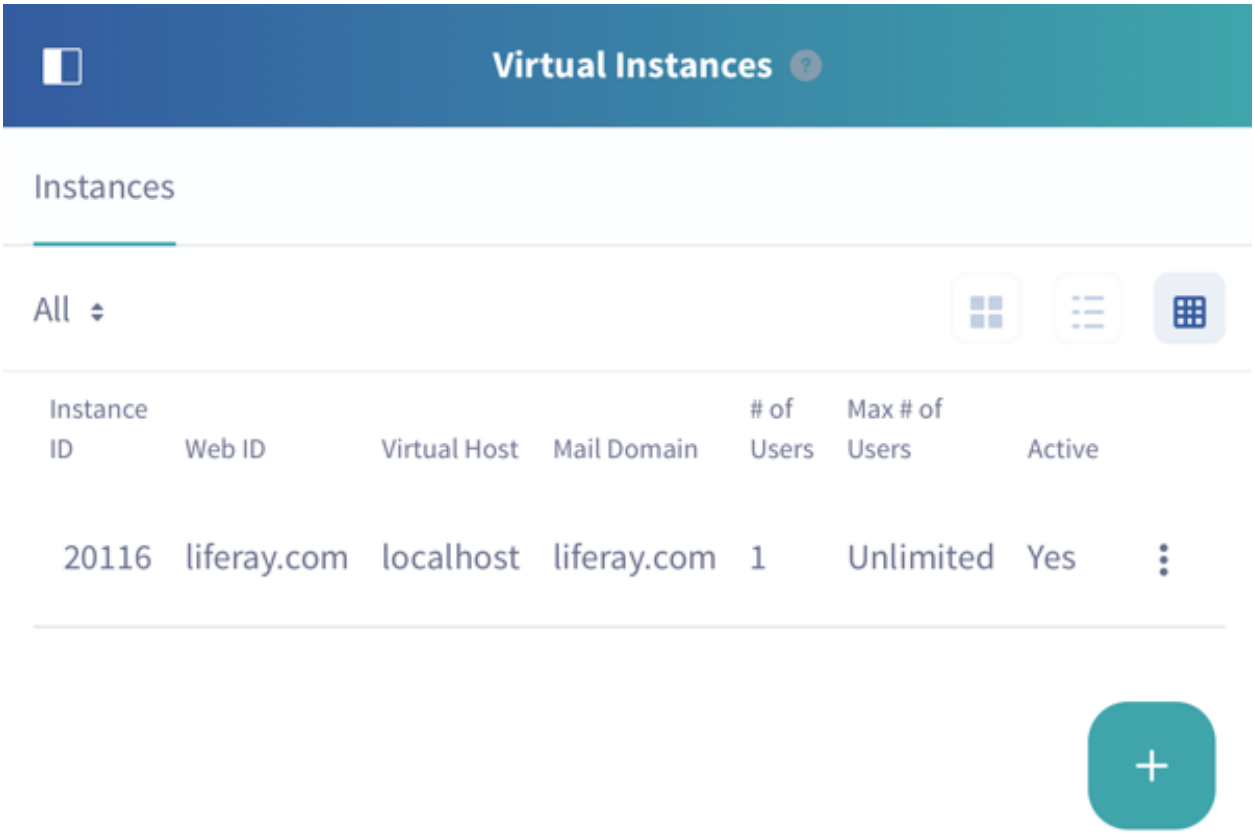


Figure 32.1: Add and manage virtual instances of Liferay in the Control Panel's *Configuration* → *Virtual Instances* section.

You'll be prompted for four fields and a check box:

Web ID: A general convention is to use the domain name for this. It's a user-generated ID for the instance.

Virtual Host: Put the domain name you configured in your network here. When users are directed to your Liferay server via this domain name, Liferay will then be able to send them to the proper virtual instance.

Mail Domain: Enter the domain name for the mail host for this instance. Liferay will use this to send email notifications from the instance.

Max Users: Enter the maximum numbers of user accounts you would like your virtual instance to support.

Active: Use this check box to choose whether to create an active or an inactive virtual instance.

When you are finished filling out the form, click *Save*. Now navigate to the instance using your new domain name. You are brought to what looks like a clean install of Liferay. This is your new virtual instance which can be configured any way you like. Read the next section on configuring an instance's settings to learn more about configuring your new instance.

32.2 Instance Settings

Most global instance settings can be configured from the Instance Settings section of the Control Panel.

Instance Settings is divided into *Configuration*, *Identification*, *Social*, and *Miscellaneous*.

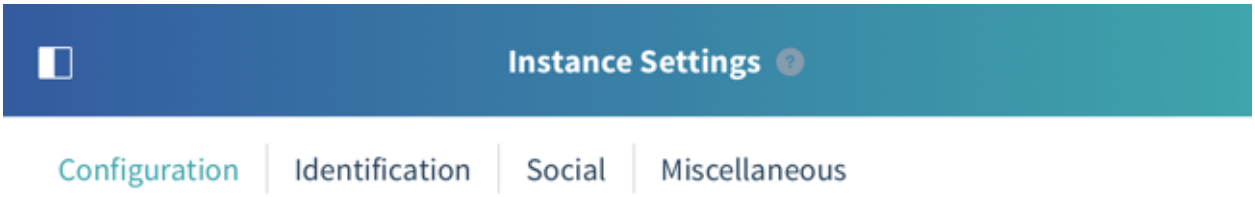


Figure 32.2: The configuration options in Instance Settings are divided into four categories.

The Configuration section is much larger than the others, so it's presented last.

Identification

The Identification section has several links for addresses, phone numbers and other information you can configure in your instance. This allows you to set up contact information for the organization that owns the instance. Developers can query for this information in their applications.

Social

The Social section lets you configure Ratings and Mentions throughout the instance. For each of Liferay's core portlets that has ratings enabled, you can choose whether to use a simple *Like*, *Thumbs*, or *Stars* for rating the content. You can also choose whether to allow users to mention (*@username*) other users, and if so, whether all users have the capability or whether you want to further refine the mentions configuration.

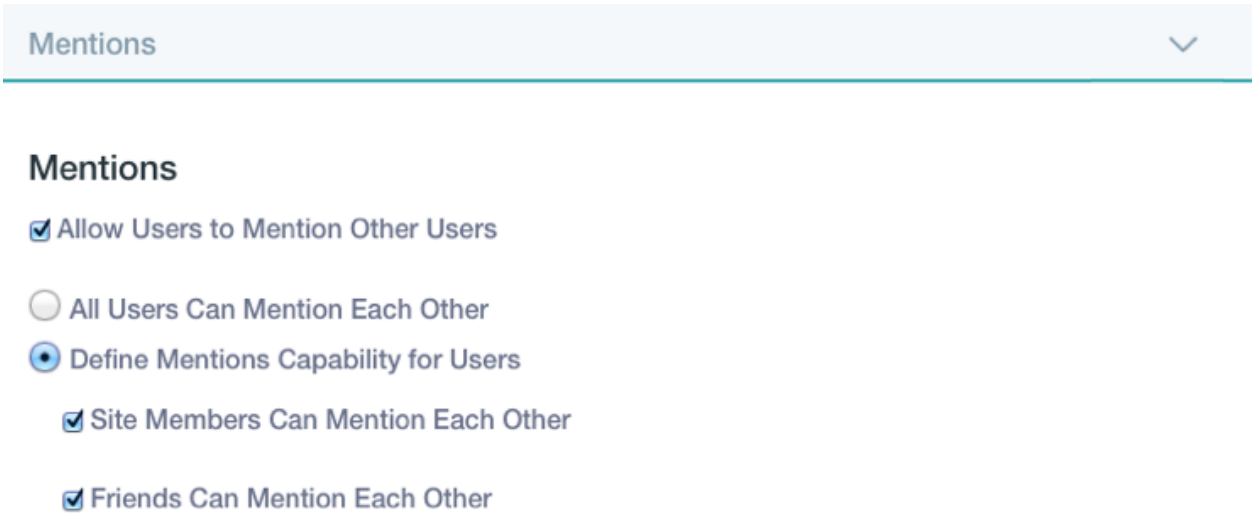


Figure 32.3: You can configure the Mentions capability throughout a Liferay instance.

Miscellaneous: Display Settings

This section allows you to set the default instance language and the time zone. You can also set up an instance-wide logo which appears in the top left corners of site pages.

Liferay's default theme is configured to display Liferay's logo. For custom themes, you can choose whether or not to display the logo. Be careful to choose an image file that fits the space. If you pick something too big, it might overlap with the navigation.

Configuration

Since there's a lot under this heading, here's a brief list, with a longer description of each following:

General: lets you configure global settings, such as the company name, domain, the virtual host, a global instance logo, and more.

Authentication: allows you to configure user authentication methods and connections to LDAP and Single Sign-On servers.

Users: has three tabs labeled Fields, Reserved Credentials and Default User Associations. The Fields tab enables or disables some user fields, such as birthday or terms of use. The Reserved Credentials tab lets you reserve screen names and email addresses so users cannot register using them. You might use this to prevent users from registering on the instance with user names that contain profanity or that sound official, such as *admin* or *president*. The Default User Associations tab lets you configure default membership to roles, user groups, sites for new users and provides a check box which allows you to retroactively apply these to existing users.

Terms of Use: decide whether to require Terms of Use, and point to a Web Content Article if you want to create your own.

Mail Host Names: lets you add a list of other mail host names to be associated with your organization. For example, your main domain might be `mycompany.com` but you might use `mycompany-marketing.com` for your email newsletters. Any domain names associated with your organization can go here.

Email Notifications: allows you to configure Liferay to send email notifications for certain events, such as user registrations, password changes, etc. You can customize those messages here.

Content Sharing: contains options for enabling site administrators to display content in one site from other sites they administer. You can also configure rules for whether subsites should be able to display content from their parent sites.

Let's discuss these settings in more detail.

General The General link takes you to a screen with three headings: Main Configuration, Navigation, and Additional Information. Under the Main Configuration heading, you can set the name of the company or organization that's responsible for running the instance. This name also defines the name of your instance's default site. Its default name is `liferay.com` so you will definitely want to change this to reflect the name of your company or organization. You can also set the mail domain, virtual host and content delivery network address here.

Under the Navigation heading, you can set a home page for your instance here as well as default landing and logout pages. For setting these pages, just use the part of the page's address that follows your domain. For example, if you want the default landing page to be `http://localhost:8080/web/guest/login`, use `/web/guest/login`. You can also use the variables `${liferay:screenName}` and `${liferay:userId}` as part of the address. This comes in handy if you want to redirect users to their personal pages upon login. Alternatively, you can set the default login or logout page in a `portal-ext.properties` file with the properties `default.landing.page.path` and `default.logout.page.path`, respectively. For more information, see the `portal.properties` documentation entries for the Default Landing Page and Default Logout Page.

Under the Additional Information heading, you can specify a Legal name, ID, company type, SIC code, ticker symbol, industry and industry type.

Authentication The Authentication section has several tabs: General, OpenSSO, LDAP, OpenID, CAS, Google, NTLM, and Facebook. You can use any of these authentication methods to configure how users will authenticate to Liferay. Since Liferay supports quite a few authentication methods, there are different settings for each.

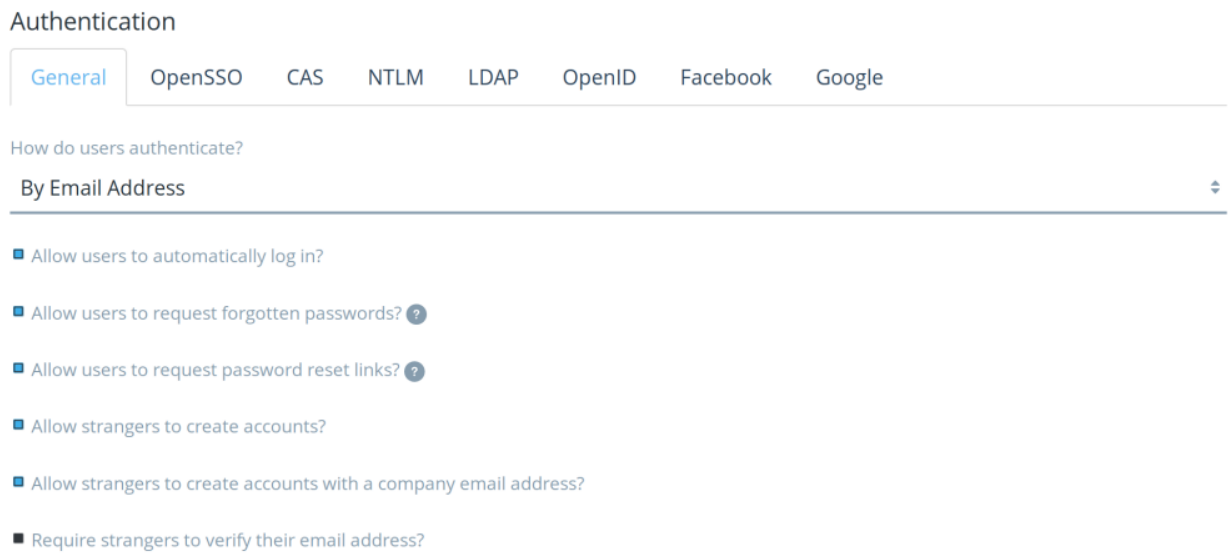


Figure 32.4: After clicking on *Instance Settings* in Liferay's Control Panel, you can configure any of the areas shown in this figure.

The settings on the General tab of the Authentication page affect only Liferay functionality and don't have anything to do with the integration options on the other tabs. The General tab allows you to customize Liferay's standard authentication behavior. Specifically, the General tab allows you to select from several global authentication settings:

- Authenticate by email address (default), screen name, or user ID (a numerical ID auto-generated in the database–not recommended).
- Enable/Disable automatic log in. If enabled, Liferay allows a user to check a box which will cause the site to "remember" the user's log in by placing a cookie on his or her browser. If disabled, users will always have to log in manually.
- Enable/Disable forgotten password functionality.
- Enable/Disable request password reset links.
- Enable/Disable account creation by strangers. If you are running an Internet site, you will probably want to leave this on so visitors can create accounts on your site.
- Enable/Disable account creation by those using an email address in the domain of the company running the site (which you just set on the General page of Instance Settings). This is handy if you are

using Liferay to host both internal and external web sites. You can make sure all internal IDs have to be created by administrators but external users can register for IDs themselves.

- Enable / Disable email address verification. If you enable this, Liferay will send users a verification email with a link back to the instance to verify the email address they entered is a valid one they can access.

By default, all settings except for the last are enabled. User authentication by email address is an important default for the following reasons:

1. An email address is, by definition, unique to the user who owns it.
2. People can generally remember their email addresses. If you have users who haven't logged into the instance for a while, it is possible they will forget their screen names, especially if they weren't allowed to use their screen names of choice (because they were already taken).
3. If a user changes his or her email address, it is more likely the user will forget to update his or her email address in his or her profile, if the email address is not used to authenticate. If the user's email address is not updated, all notifications sent by the instance will fail to reach the user. So it is important to keep the email address at the forefront of a user's mind when he or she logs in to help the user keep it up to date.

If you need to use a third party authentication service or set up Single Sign On (SSO), see the documentation on Liferay DXP Security to learn about the supported services and how to configure them.

Users If you click on *Users* from the Instance Settings screen, you'll find three tabs: Fields, Reserved Credentials and Default User Associations.

The Fields tab allows you to enable/disable these fields on the Add/Edit User Form:

- Autogeneration of screen names
- Birthday field
- Gender field

In the Reserved Credentials tab, enter any screen names and email addresses that you don't want users to claim. Liferay prevents users from registering with these screen names and email addresses. You might use this feature to prevent users from creating IDs that look like administrative IDs or that have reserved words in their names.

The Default User Associations tab has three fields allowing you to list (one per line) sites, organization sites, roles, and user groups you want new users to become members of automatically. By default, Liferay assigns new users to both the Users role and the Power Users role. You can also choose whether to apply these configurations to existing users.

User groups are handy tools for pre-populating your users' private sites with a certain page set, defined by a site template. If you've done that and want all users you add to immediately be added to the user group and inherit those pages, add the user group here.

Terms of Use Do you want to disable the requirement for all users to read the terms of use? You can do that, or even specify the Group ID and Article ID for a Web Content Article that you'd like to use as your own custom Terms of Use.

Mail Host Names You can enter other mail host names (one per line) besides the one you configured on the General tab. This lets the instance know which mail host names are owned by your organization.

Email Notifications There are five tabs under the Email Notifications page of Instance Settings. The Sender tab allows you to set the instance's administrative name and email address. The Email Sender entry specifies the virtual instance's administrative Name and Address for email notifications, declared as the `[$FROM_NAME$]` and `[$FROM_ADDRESS$]` variables respectively in the email templates. By default, they are from the `admin.email.from.name` and `admin.email.from.address` portal properties. This name and email address appear in the *From* field in all email messages sent by the virtual instance.

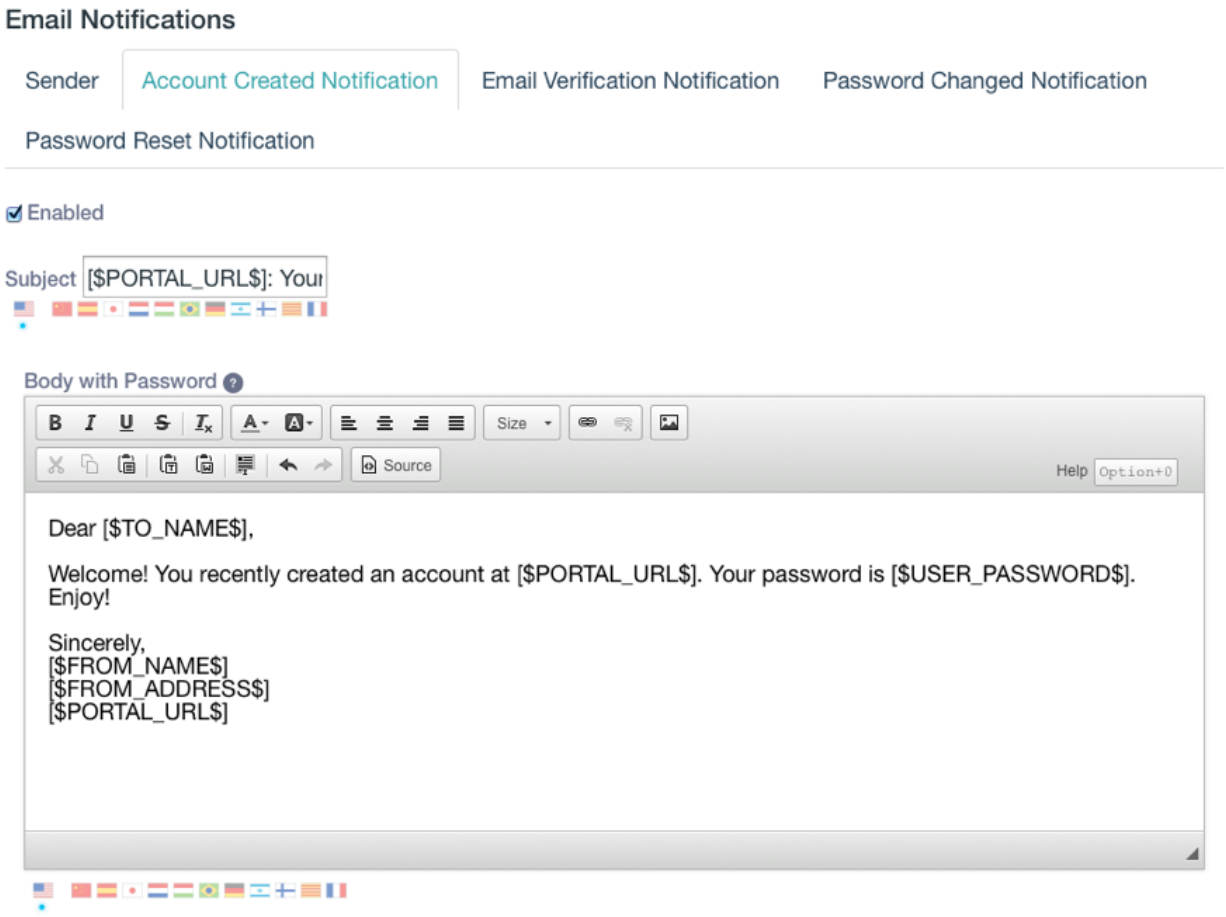


Figure 32.5: You can customize the email template for the email messages sent to users who have just created their accounts.

The other four tabs are Account Created Notification, Email Verification Notification, Password Changed Notification and Password Reset Notification. These tabs allow you to customize the email messages that are sent to users each time any of those four events occur.

A list of tokens, entitled "Definition of Terms," is provided so you can insert certain values (such as the portal URL or the user ID) when you are setting up the custom email messages.

Content Sharing The Content Sharing section of the Instance Settings area of the Control Panel allows you to choose whether or or not site administrators can display content in sites from other sites they administer. For example, suppose that a certain user is a site administrator of two sites: *Engineering* and

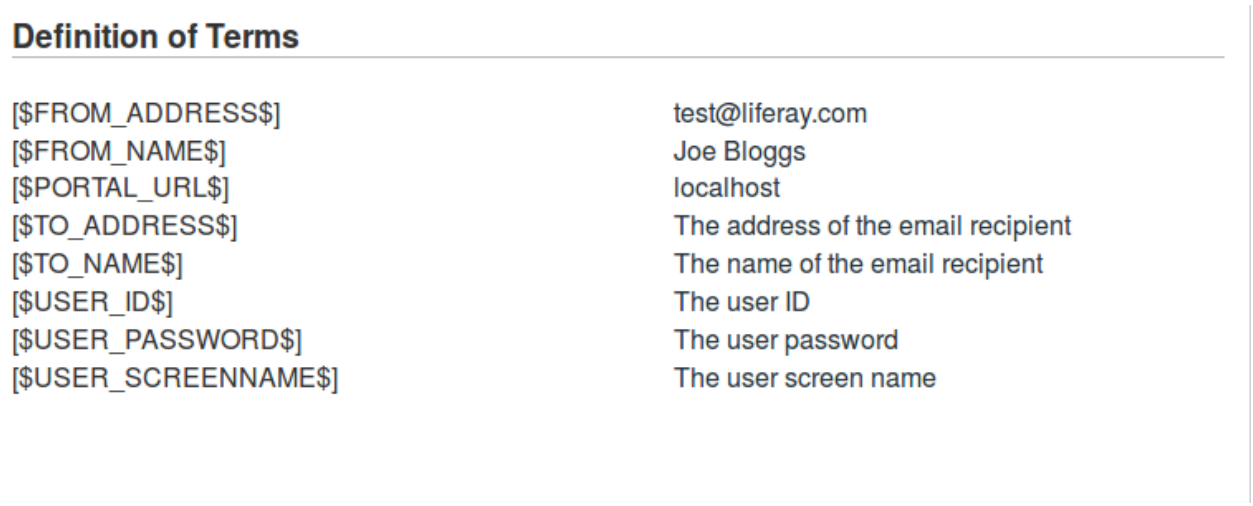


Figure 32.6: There are some handy variables available for use in email templates.

Marketing. The checkbox in the Content Sharing section of Instance Settings determines whether or not the site administrator can display content from the Marketing site in the Engineering site and vice versa.

The Content Sharing section of Instance Settings also allows you to choose a behavior for whether or not subsites can display content from parent sites and for configuring the defaults. There are three options:

Enabled by Default: This means that subsites can display content from parent sites by default but this can be disabled by a site administrator.

Disabled by Default: This means that subsites cannot display content from parent sites by default but this can be enabled by a site administrator.

Disabled: This means that subsites cannot display content from parent sites and this behavior cannot be changed by a site administrator.

Next, learn to integrate existing users from other environments, such as LDAP servers, into Liferay.

CHAPTER 33

USING LIFERAY'S SCRIPT ENGINE

Liferay DXP provides a robust script engine for executing Groovy scripts to maintain your Liferay DXP instance. You can execute scripts to perform maintenance tasks involving data cleanup, user maintenance operations, bulk Liferay API invocations, or even system level operations.

The tutorials in this section explain how to use Liferay's script engine and script console. They cover the following topics:

- Invoking Liferay's API from a script
- Running scripts from the script console
- Using the script engine with workflow
- Leveraging custom Java tools in the script engine

Invoking Liferay's API is probably the most common task for which you'll want to run a script. If you have any familiarity with Liferay's API, this will be very easy for you. To access Liferay's scripting console, navigate to the Control Panel, click on *Server Administration* under the System heading, then click on the *Script* tab.

33.1 Invoking Liferay Services From Scripts

Many scripting scenarios require invoking Liferay DXP's services.

To illustrate the correct syntax for interacting with Liferay services, consider a simple example that uses the `UserLocalService` API to retrieve a list of users and print their names to Liferay's log file. We'll initially implement the example in Java pseudo-code:

```
import com.liferay.portal.kernel.model.User;
import com.liferay.portal.kernel.service.UserLocalServiceUtil;
import java.util.List;

...

int userCount = UserLocalServiceUtil.getUsersCount();
List<User> users = UserLocalServiceUtil.getUsers(0, userCount);

for (User user:users) {
    System.out.println("User Name: " + user.getFullName());
```

```
}

...
```

Liferay DXP's script engine only supports Groovy by default. In later versions, support may be added for other scripting languages.

Groovy

Groovy is based on Java, and code written in Java also runs in Groovy. This means we can execute the exact same code from our Java example without any changes:

```
import com.liferay.portal.kernel.model.User;
import com.liferay.portal.kernel.service.UserLocalServiceUtil;
import java.util.List;

int userCount = UserLocalServiceUtil.getUsersCount();
List<User> users = UserLocalServiceUtil.getUsers(0, userCount);

for (User user:users) {
    System.out.println("User Name: " + user.getFullName());
}
```

Of course, we could make this somewhat Groovier by simplifying the program as follows:

```
import com.liferay.portal.kernel.service.UserLocalServiceUtil

userCount = UserLocalServiceUtil.getUsersCount()
users = UserLocalServiceUtil.getUsers(0, userCount)
for (user in users){
    System.out.println("User Name: " + user.getFullName())
}
```

Liferay's services can be easily accessed from the script console. Next, let's look at some practical uses for Liferay DXP's script engine.

Related Topics

Running Scripts From the Script Console
Leveraging the Script Engine in Workflow
Using Liferay's Script Engine

33.2 Running Scripts From the Script Console

To see a very simple example of the script console in action, log into the portal as an administrator and navigate to the *Control Panel → Server Administration* area of the Control Panel. Then click on *Script*. This is Liferay DXP's script console. Change the script type to *Groovy* and replace the code in the script console with the following:

```
number = com.liferay.portal.kernel.service.UserLocalServiceUtil.getUsersCount();
out.println(number);
```

Click the *Execute* button and check the script console or the log for the output.

Next, consider a less simplistic example. You'll retrieve some user information from the database, make some changes, and then save those changes to Liferay's database. Suppose that your company has updated the terms of use and wants each user to be presented with the updated terms of use whenever they next log

in. When users agree to the terms of use, a boolean attribute called `agreedToTermsOfUse` is set in their user records. As long as the value of this variable is `true`, Liferay DXP will not present the user with the terms of use. However, if you set this flag to `false` for each user, each user must agree to the terms of use again before they can log in.

You'll use Groovy again. Ensure that the script type in the script console is set to Groovy. Then execute the following code to check the status of the `agreedToTermsOfUse` user attribute:

```
import com.liferay.portal.kernel.service.UserLocalServiceUtil

userCount = UserLocalServiceUtil.getUsersCount()
users = UserLocalServiceUtil.getUsers(0, userCount)

for (user in users) { println("User Name: " + user.getFullName() + " -- " +
user.getAgreedToTermsOfUse()) }
```

The code above just prints the value of the `agreedToTermsOfUse` attribute for each user. Next, you'll actually update each user in the system to set his or her `agreedToTermsOfUse` attribute to `false`. Your script will make sure to skip the default user as well as the default admin user that's currently signed in and running the script.

```
import com.liferay.portal.kernel.service.UserLocalServiceUtil

userCount = UserLocalServiceUtil.getUsersCount()
users = UserLocalServiceUtil.getUsers(0, userCount)

long currentUserId = Long.parseLong(userInfo.get("liferay.user.id"))

for (user in users){

    if(!user.isDefaultUser() && (user.getUserId() ≠ currentUserId)) {

            user.setAgreedToTermsOfUse(false)
            UserLocalServiceUtil.updateUser(user)

    }

}
```

To verify the script has updated the records, run the first script again and you should see that all users (except the default user and your user) have been updated.

That's all that's needed to run scripts and to access the Liferay service layer.

Keep these things in mind when working with the script console:

- There is no undo.
- There is no preview.
- When invoking local services, no permissions checking is enforced.
- Scripts are executed synchronously. Be careful with scripts that might take a long time to execute.

For these reasons, you should use the script console cautiously. It's best to test run your scripts on non-production systems before running them on production. Follow the tips in the subsequent sections to make better use of Liferay's script console. Note: These tips originated from a Liferay blog post. The following scripts are Groovy scripts but they can be adapted to other languages.

Tip 1: Use the Predefined Variables

The following predefined variables are available to scripts executed from Liferay's script console:

- `out` (`java.io.PrintWriter`)
- `actionRequest` (`javax.portlet.ActionRequest`)
- `actionResponse` (`javax.portlet.ActionReponse`)
- `portletConfig` (`javax.portlet.PortletConfig`)
- `portletContext` (`javax.portlet.PortletContext`)
- `preferences` (`javax.portlet.PortletPreferences`)
- `userInfo` (`java.util.Map<String, String>`)

Note that if you use `System.out.println`, for example, your output will be printed to Liferay's log file. If you use `out.println` instead (using the predefined variable), your output will be printed to the script console.

The predefined variables can all be very useful when you're creating scripts. The `actionRequest` variable can be especially useful, as this script demonstrates:

```
import com.liferay.portal.kernel.util.*

company = PortalUtil.getCompany(actionRequest)
out.println("Current Company:${company.getName()}\n")

out.println("User Info:")
userInfo.each {
        k,v -> out.println("${k}:${v}")
}
```

Tip 2: Embed HTML Markup in Script Outputs

The output of the script console is rendered as HTML content. Thus, you can embed HTML markup in your outputs to change their look and feel. Here's an example:

```
import com.liferay.portal.kernel.service.*

number = com.liferay.portal.kernel.service.UserLocalServiceUtil.getUsersCount();
out.println(
        """
                <div style="background-color:black; text-align: center">
                        <h1 style="color: #37A9CC; font-size:xx-large">${number}</h1>
                </div>
        """);
```

Tip 3: Show Exceptions in the Script Console

When any exception occurs during script execution, the error message is always the same:

```
Your request failed to complete.
```

This message gives no detail about the error. To find information about the error and what caused it, you usually need to examine the server logs.

However, you can use the following technique to make exception details appear in the script console. Wrap your code with a try / catch block and print the stack trace to the console output from the catch clause. Note that even this technique does not catch script syntax errors. Here's an example:

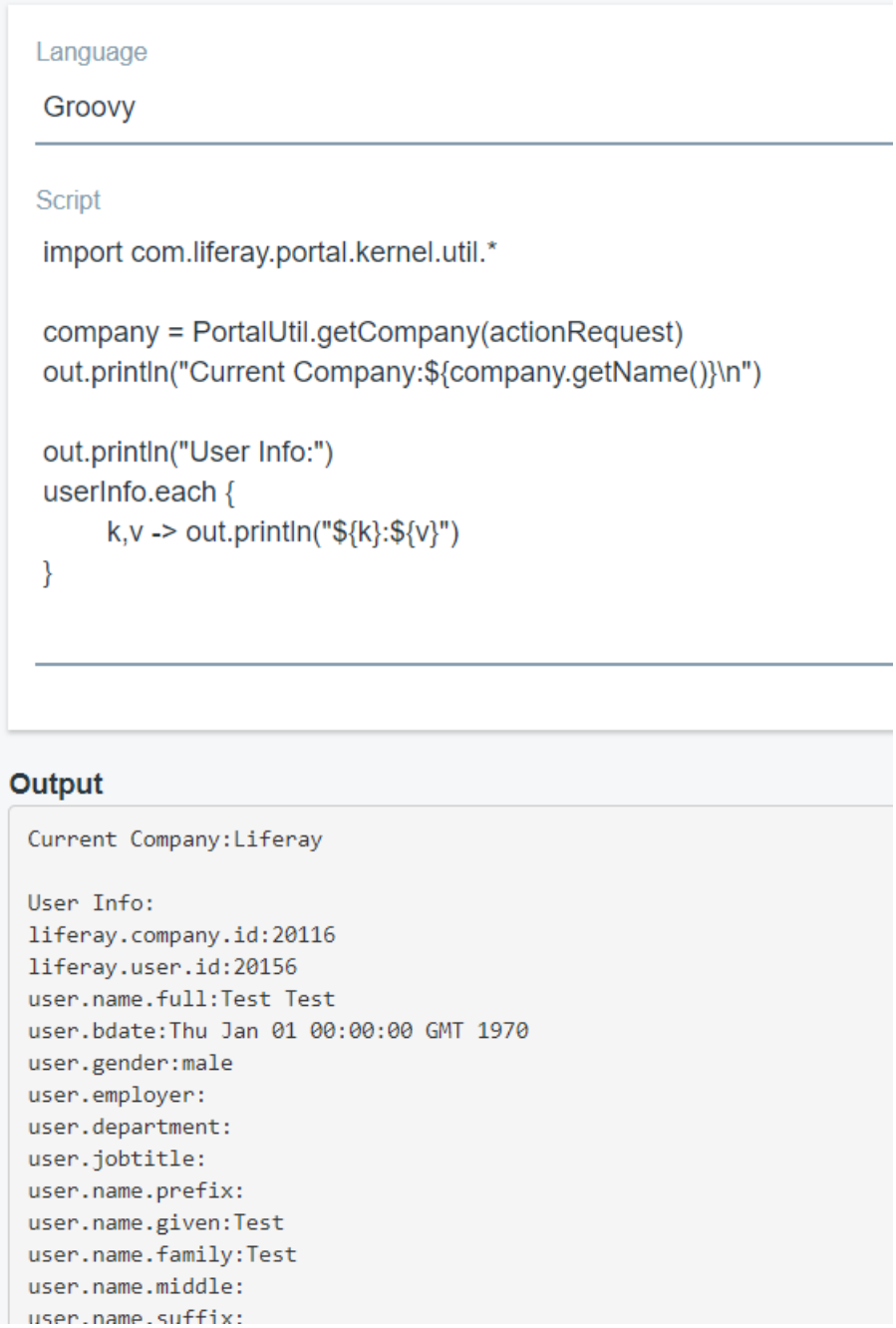


Figure 33.1: Here's an example of invoking a Groovy script that uses the predefined `out`, `actionRequest`, and `userInfo` variables to print information about the current user.

Figure 33.2: Here's an example of invoking a Groovy script that embeds HTML markup in the output of the script.

```
try {
        nullVar = null
        out.println(nullVar.length())
} catch(e) {
        out.println("""<div class="portlet-msg-error">${e}</div>""")
        e.printStackTrace(out)
}
```

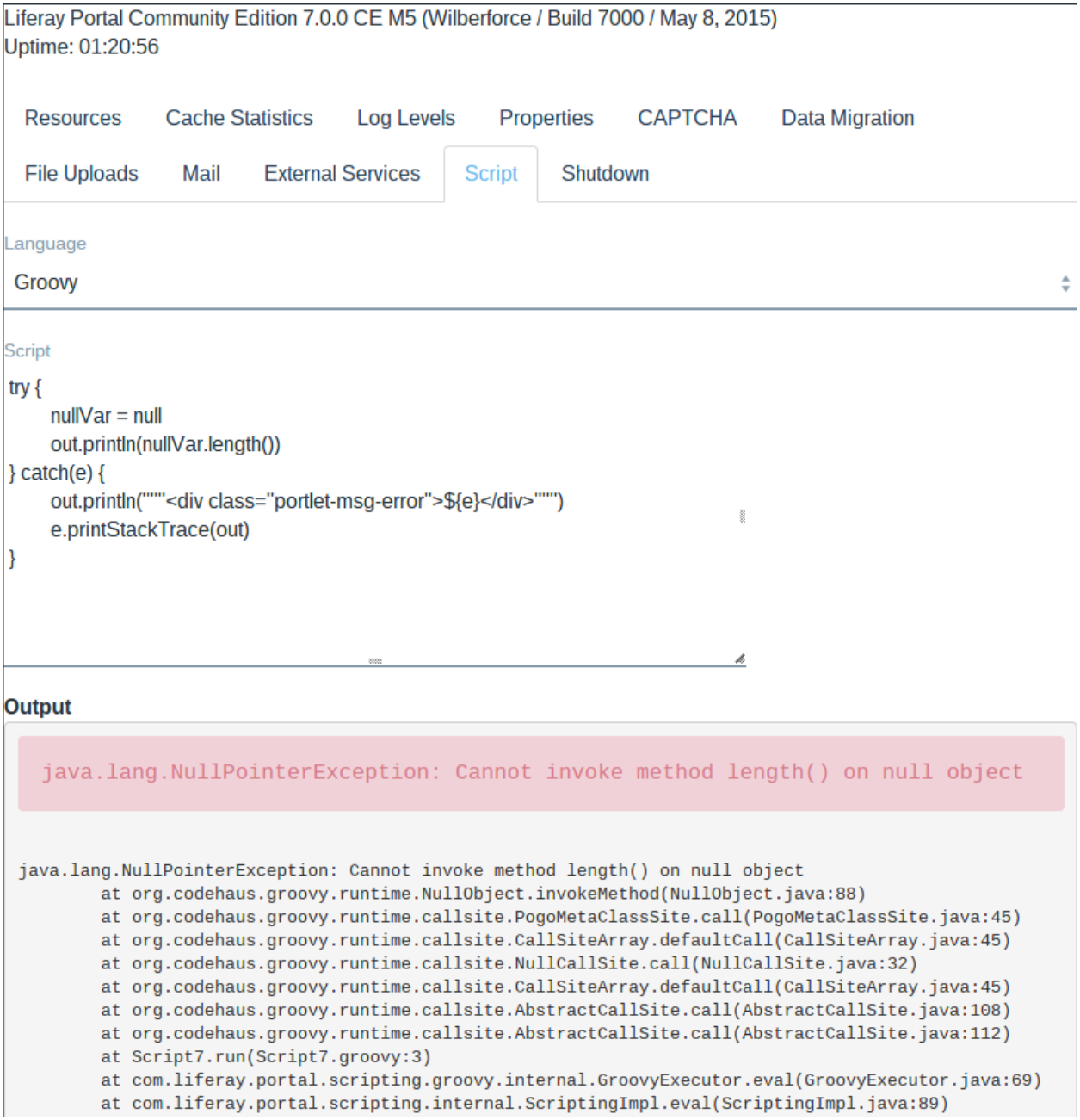


Figure 33.3: Here's an example of a Groovy script that catches exceptions and prints exception information to the script console.

Tip 4: Implement a Preview Mode

Since Liferay's script console does not provide an undo feature, it can be very convenient to set up a kind of preview mode. The purpose of a preview mode is to determine any permanent effects of a script before any information is actually saved to the Liferay database. The preview mode consists in using a `previewMode` flag which determines whether the operations with permanent effects should be executed or not. If `previewMode` is true, all of the data which would be permanently affected by the script is systematically printed. This providers the user executing the script with an outline of the data impacted by the script. If the user determines that everything is OK, the flag can be switched so that the script can make permanent updates to the database.

Here's an example Groovy script that show the preview mode concept in action:

```
import java.util.Calendar
import com.liferay.portal.kernel.service.*
import com.liferay.portal.kernel.model.*
import com.liferay.portal.kernel.dao.orm.*
import static com.liferay.portal.kernel.workflow.WorkflowConstants.*

//
// Deactivate users never logged and created since more than 2 years
//

previewMode = true // Update this flag to false to really make changes

Calendar twoYearsAgo = Calendar.getInstance()
twoYearsAgo.setTime(new Date())
twoYearsAgo.add(Calendar.YEAR, -2)

DynamicQuery query = DynamicQueryFactoryUtil.forClass(User.class)
        .add(PropertyFactoryUtil.forName("lastLoginDate").isNull())
        .add(PropertyFactoryUtil.forName("createDate").lt(twoYearsAgo.getTime()))

users = UserLocalServiceUtil.dynamicQuery(query)

users.each { u ->
         if(!u.isDefaultUser() && u.getStatus() != STATUS_INACTIVE) {
                out.println(u.getEmailAddress())
                if(!previewMode) {
                        UserLocalServiceUtil.updateStatus(u.getUserId(), STATUS_INACTIVE)
                }
         }
}

if(previewMode) {
        out.println('Preview mode is on: switch off the flag and execute '
                + 'again this script to make changes to the database')
}
```

Tip 5: Plan a File Output for Long-Running Scripts

When a script has been running for a long time, it's possible for the script console to return an error even though the script can continue running and potentially conclude successfully. But it's impossible to know the outcome without the corresponding output!

To bypass this limitation, you can send the output of the script console to a custom file instead of to the console itself or to the Liferay log. For example, consider this script:

```
import com.liferay.portal.kernel.service.*
import com.liferay.portal.kernel.dao.orm.*

// Output management
```

```
final def SCRIPT_ID = "MYSCRIPT"
outputFile = new File("""${System.getProperty("liferay.home")}/scripting/out-${SCRIPT_ID}.txt""")
outputFile.getParentFile().mkdirs()

def trace(message) {
        out.println(message)
        outputFile << "${message}\n"
}

// Main code

users = UserLocalServiceUtil.getUsers(QueryUtil.ALL_POS, QueryUtil.ALL_POS)
users.each { u ->
        trace(u.getFullName())
}
```

The script above creates a subfolder of Liferay Home called `scripting` and saves the script output to a file in this folder. After running the script above, it's possible to read the generated file without direct access to the file system. Here's a second script that demonstrates this:

```
final def SCRIPT_ID = "MYSCRIPT"
outputFile = new File("""${System.getProperty("liferay.home")}/scripting/out-${SCRIPT_ID}.txt""")
out.println(outputFile.text)
```

One advantage of using a dedicated output file instead of using a classic logger is that it's easier to get the script output data back. Getting the script output data would be more difficult to obtain from the portal log, for example, because of all the other information there.

Of course, Liferay's script engine has uses beyond the script console. Next, you'll learn how to leverage Liferay's script engine for designing workflows.

Related Topics

Invoking Liferay Services From Scripts

Leveraging the Script Engine in Workflow

Using Liferay's Script Engine

33.3 Leveraging the Script Engine in Workflow

Liferay's Kaleo workflow engine provides a robust system for reviewing and approving content in an enterprise environment. Even if you don't leverage custom scripts, it's a powerful and robust workflow solution. Adding custom scripts takes it to the next level.

The final step in a workflow should run a script that makes content available for use. As you can see in the snippet below, JavaScript can be used to access the Java class associated with the workflow to set the status of the content to *approved*.

```
<script>
    <![CDATA[
        Packages.com.liferay.portal.kernel.workflow.WorkflowStatusManagerUtil.updateStatus(Packages.com.liferay.portal.kernel.workflow.WorkflowConstants.toS
        Packages.com.liferay.portal.kernel.workflow.WorkflowStatusManagerUtil.updateStatus(Packages.com.liferay.portal.kernel.workflow.WorkflowConstants.toS
    ]]>
</script>
<script-language>javascript</script-language>
```

At virtually any point in a workflow, you can use Liferay's script engine to access workflow APIs or other Liferay APIs. There are a lot of different ways you could use this. Here are a few practical examples:

- Getting a list of users with a specific workflow-related role
- Sending an email to the designated content approver with a list of people to contact if he is unable to review the content
- Creating an alert to be displayed in the Alerts portlet for any user assigned to approve content

---

Note: In earlier versions of Liferay DXP, a bug made it difficult (without a messy workaround) to resolve classes outside of Liferay DXP's core (for example, the `portal-kernel` module). That bug is fixed as of the following versions:

- Liferay Digital Enterprise 7.0 Fix Pack 11
- Liferay Portal 7.0 GA 4

If your workflow script depends on services in a module outside of Liferay DXP's core (for example, `JournalArticleLocalService`), make sure you are running an up-to-date version of Liferay DXP.

Of course, before you try any of this, you need to know the appropriate syntax for inserting a script into a workflow. In an XML workflow definition, a script can be used in any XML type that can contain an *actions* tag: those types are `<state>`, `<task>`, `<fork>` and `<join>`. Inside of one of those types, format your script like this:

```
<actions>
    <action>
        <script>
            <![CDATA[*the contents of your script*]]>
        </script>
        <script-language>*your scripting language of choice*</script-language>
    </action>
    ...
</actions>
```

Here's an example of a workflow script created in Groovy. This one is designed to be used with a `Condition` statement in Kaleo. It accesses Liferay's asset framework to determine the category of an asset in the workflow. The script uses the category to automatically determine the correct approval process. If the category `legal` has been applied to the asset, the asset is sent to the `Legal Review` task upon submission. Otherwise, the asset is sent to the `Default Review` task.

```
<script>
    <![CDATA[
        import com.liferay.portal.kernel.util.GetterUtil;
        import com.liferay.portal.kernel.workflow.WorkflowConstants;
        import com.liferay.portal.kernel.workflow.WorkflowHandler;
        import com.liferay.portal.kernel.workflow.WorkflowHandlerRegistryUtil;
        import com.liferay.portlet.asset.model.AssetCategory;
        import com.liferay.portlet.asset.model.AssetEntry;
        import com.liferay.portlet.asset.model.AssetRenderer;
        import com.liferay.portlet.asset.model.AssetRendererFactory;
        import com.liferay.portlet.asset.service.AssetEntryLocalServiceUtil;

        import java.util.List;

        String className = (String)workflowContext.get(
            WorkflowConstants.CONTEXT_ENTRY_CLASS_NAME);

        WorkflowHandler workflowHandler =
            WorkflowHandlerRegistryUtil.getWorkflowHandler(className);

        AssetRendererFactory assetRendererFactory =
```

```
            workflowHandler.getAssetRendererFactory();

        long classPK =
            GetterUtil.getLong((String)workflowContext.get
            (WorkflowConstants.CONTEXT_ENTRY_CLASS_PK));

        AssetRenderer assetRenderer =
            workflowHandler.getAssetRenderer(classPK);

        AssetEntry assetEntry = assetRendererFactory.getAssetEntry(
            assetRendererFactory.getClassName(), assetRenderer.getClassPK());

        List<AssetCategory> assetCategories = assetEntry.getCategories();

        returnValue = "Default Review";

        for (AssetCategory assetCategory : assetCategories) {
            String categoryName = assetCategory.getName();

            if (categoryName.equals("legal")) {
                returnValue = "Legal Review";

                return;
            }
        }
       ]]>
</script>
<script-language>groovy</script-language>
```

Within a workflow, the next task or state is chosen based on the return value. For a complete example of a workflow script that uses the above Groovy script, please see this `legal-workflow-script.xml` file: https://github.com/liferay/liferay-docs/blob/6.2.x/userGuide/code/legal-workflow-script.xml.

Calling OSGi Services

How do you call OSGi services from a workflow script, accounting for the dynamic environment of the OSGi runtime, where services your script depends on can disappear without notice? Use a service tracker. That way you can check to make sure your code has access to the service it needs, and if not, do something appropriate in response. Here's a little example code to show you how this might look in Groovy (import statements excluded):

```
ServiceTracker<SomeLocalService,SomeLocalService> st;

try {
    Bundle bundle = FrameworkUtil.getBundle(GroovyExecutor.class);

    st = new ServiceTracker(bundle.getBundleContext(), JournalArticleLocalService.class, null);
    st.open();

    if (!st.isEmpty()) {
        SomeLocalService _SomeLocalService = st.getService();

        //Do cool stuff with the service you retrieved
    }
}
catch(Exception e) {
    //Handle error appropriately
}
finally {
    if (st ≠ null) {
        st.close();
    }
}
```

If you read the article on service trackers, the only odd looking piece of the above code is the `getBundle` call: why is `GroovyExecutor.class` passed as a parameter? The parameter passed to the `FrameworkUtil.getBundle` call must be a class from the bundle executing the workflow script. This is different from the context of a plugin project, where you'd want to get the bundle hosting the class where you're making the call (using `this.getClass()`, for example). Note that for another scripting engine, you must pass in a concrete class from the particular bundle executing your script.

The combination of Liferay's script and workflow engines is incredibly powerful. However, since it provides users with the ability to execute code, it can be dangerous. When configuring your permissions, be aware of the potential consequences of poorly or maliciously written scripts inside of a workflow definition. For more information on creating workflow definitions with Kaleo workflow, see Liferay DXP's workflow documentation.

Related Topics

Invoking Liferay Services From Scripts

Running Scripts From the Script Console

Using Liferay's Script Engine

33.4 Using Web Services for Remote Portlets (WSRP)

The Web Services for Remote Portlets (WSRP) specification defines a web service interface for accessing and interacting with presentation-oriented web services in the form of portlets. What are presentation-oriented web services? These are web services that send user interfaces over the wire, rather than raw data like JSON objects or SOAP data envelopes. If an application is written as a portlet, this is an easy way to expose that application to end users on a completely different system, rather than sending just the data and having to craft an application to present that data. WSRP's presentation-oriented web services allow portals to display remote portlets inside their pages, as if locally deployed, without requiring any additional programming by developers.

Here are the two main components for WSRP:

Producer: A web service that exposes one or more portlets and is described using a Web Services Description Language (WSDL) document.

Consumer: A web service client that receives the data from the Producer and presents it to the user in a portlet window.

Warning: The WSRP portlet is not recommended for using on the Internet because encryption between the producer and consumer is not supported.

Below, you'll see how the components interact with each other. So without further ado, let's explore WSRP in Liferay DXP!

WSRP with Liferay

Liferay DXP provides a deployable WSRP portlet that supports the 1.0 and 2.0 specifications. The portlet is available from Liferay Marketplace as a CE or EE app. Once you've downloaded and installed the WSRP app, you have instant access to the portlet by navigating to the Control Panel and, under *Configuration*, selecting *WSRP Admin*.

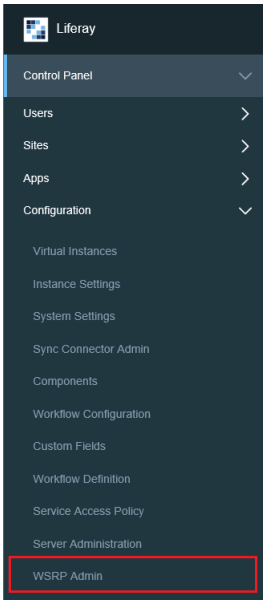


Figure 33.4: Once deployed, you can find the WSRP Admin portlet in the Control Panel.

Liferay DXP can be used as a WSRP producer or consumer. As a producer, it hosts portlets that are consumed by other portal servers (Liferay or non-Liferay) acting as WSRP consumers. The image below illustrates WSRP producers and consumers and how they interact.

Let's go through the basic process of how producers and consumers work together to bring the end user a remote portlet. First, the consumer portal server establishes a connection with its producer portal server counterpart. This connection is made possible by giving the consumer the producer web service's URL. The consumer then uses the URL to discover the producer's portlets and establish a connection. After the connection is made, the consumer acquires the producer's information and creates a consumer proxy portlet. The proxy portlet acts as an intermediary, relaying requests to and from the end user and the producer's portlet.

For example, you can compare the proxy portlet to a TV satellite box. If you want to change the channel on your TV, you (end user) send the channel number you desire to the TV's satellite box (consumer's proxy portlet) via your TV's remote. When the satellite box receives the request to change the channel, it relays the request to a TV satellite (producer's portlet) which then sends the channel information back to the satellite box. Then, the satellite box displays the new channel to you on your TV. In this simple example, you're not directly requesting the TV satellite to change the channel, but rather, you're communicating with the satellite box, which acts as an intermediary between you and the satellite. This example directly relates to using WSRP with Liferay. Although the end users are sending requests to the consumer portlet, they're not receiving feedback from the consumer portlet itself, but rather its producer portlet located remotely.

Now that you know a little bit about the WSRP process, let's begin configuring WSRP on Liferay DXP. For this demonstration, you'll assume you have two portal servers.

Tip: If you're following along with this example and don't have an additional portal server, you can download another instance of Liferay DXP and have it running at the same time as your current Liferay instance to simulate an additional portal server. Remember, typical use cases have WSRP producers and consumers linked on differing portal servers. To run two portal instances locally at the same time, you'll need to change one of your portal's server configurations. Make sure your portals use different ports. If you're running Tomcat servers, for example, open their `tomcat-[VERSION]/conf/server.xml` files and make sure their `port=`

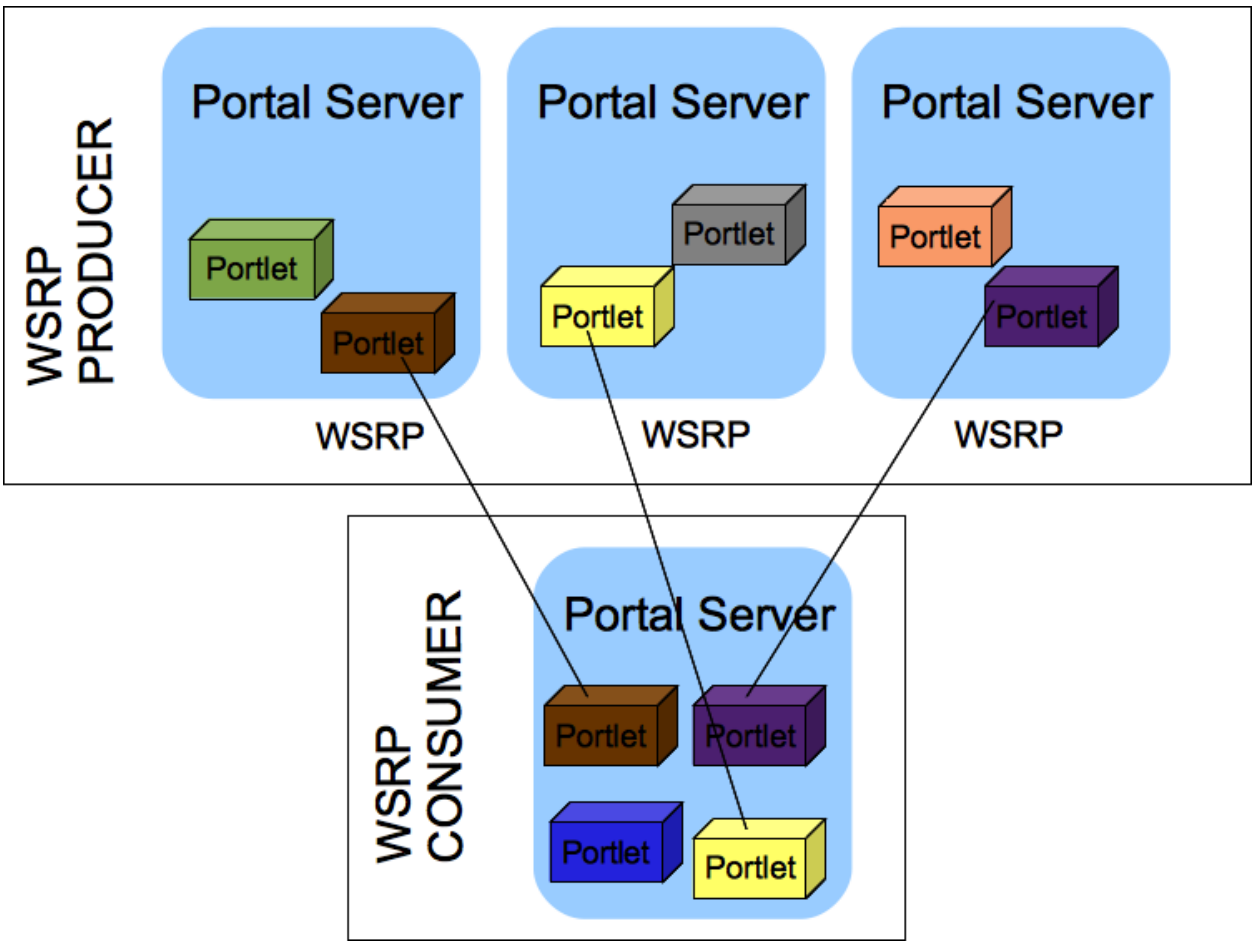


Figure 33.5: Portlets can interact with other portlets located on a different portal server using WSRP.

designations use different values (e.g., change one server's port from `8080` to `18080`). Also, you can specify the new port number for your browser launcher URL by adding `browser.launcher.url=http://localhost:18080` to modified portal's `portal-ext.properties` file.

To create a producer, go to the *Producers* tab and click the *Add* icon ( ). Name your producer `Hello World Producer` and choose the appropriate version of WSRP to use. Liferay displays a list of available portlets your producer can use. For demonstration purposes, select the Hello World portlet and click the *Save* button. The portal generates a WSDL document to define your producer. You can copy the WSDL document URL for accessing the producer in a consumer.

Now that you've created a producer, let's create a consumer on your second portal server.

On your consumer portal server, navigate to the *WSRP Admin* page. Select the *Consumers* tab and click the *Add* icon ( ). Give the consumer a name and add the producer's WSDL URL in the *URL* field. There are also additional fields:

Forward Cookies: Allows the WSRP consumer to forward specific cookies from the user's browser session to the WSRP producer.

Forward Headers: Allows the WSRP consumer to forward specific HTTP headers from the user's browser session to the WSRP producer.

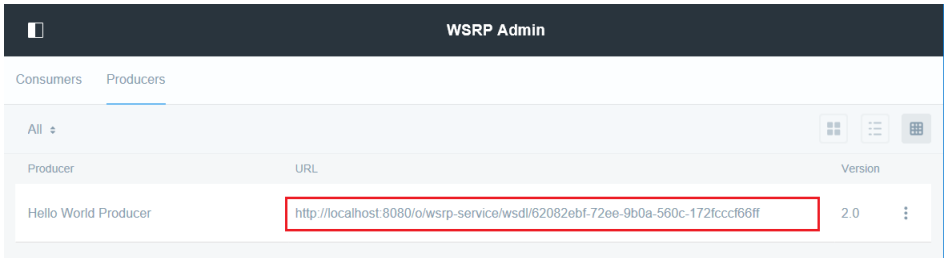


Figure 33.6: You can copy the WSDL document URL for using the producer's portlets in a consumer.

Markup Character Sets: Markup character encodings supported for the consumer are shown in a comma delimited list. UTF-8 is assumed and is added automatically as a supported encoding.

Leave these additional fields blank for our demonstration. Lastly, you need to define the portlets that the end-user can use from this consumer. To do this, click the consumer's *Actions* icon ( ) and select *Manage Portlets*. The Manage Portlets screen appears.

The producer's remote portlets are available to add to your consumer. To add the *Hello World* remote portlet, click the *Add* icon ( ) The New Portlet screen appears. Select the *Hello World* remote portlet in the Remote Portlet field and give the consumer's new portlet an arbitrary name. Then click *Save*. Now end users can "consume" or use the remote portlet just like any local portlet in the portal.

Next, you'll learn how to create custom remote portlets.

Creating Custom Remote Portlets

With the demand for dynamic portlets by end users, sometimes a finite, pre-selected list of remote portlets isn't enough. Because of this, Liferay DXP allows you to make custom developed portlets remotely accessible for WSRP.

To enable your custom portlet for WSRP, add the `<remoteable>true</remoteable>` tag in your portlet's `docroot/WEB-INF/liferay-portlet.xml` file:

```
<liferay-portlet-app>
    <portlet>
        <portlet-name>RemoteSamplePortlet</portlet-name>
        <remoteable>true</remoteable>
...
    </portlet>
...
</liferay-portlet-app>
```

After editing your portlet's `liferay-portlet.xml` file, your custom portlet appears in the list of portlets available to WSRP producers. Congratulations! Now you can share all your portlets to end users using WSRP!

Next, you'll learn how to remotely access Liferay services.

33.5 Configuring Form Navigator Forms

Liferay DXP's forms can be built to gather large amounts of data. These larger forms are organized into easily navigable sections and categories. This is possible thanks to the Form Navigator framework. As an administrator, you may wish to hide certain form sections. Maybe you want to show certain sections when creating new content, but hide other sections when someone updates that content. Perhaps you want to

reorder form sections. Since Liferay DXP 7.0 Fix Pack 14 and Liferay Portal CE GA4, all these configuration options are possible via the *Form Navigator Configuration* system setting.

This tutorial covers how to use the *Form Navigator Configuration* system setting to control which form categories and sections are visible.

Note: Individual fields within form sections cannot be hidden using this method. Your developers can use a Core JSP Hook to override a form's JSP if you need to customize the individual fields within a form section.

Get started by retrieving the available entry keys and categories next.

Running the Form Navigator Gogo Shell Command

Form Navigator configurations contain the following information:

- Form ID
- Category (if there is one)
- Context (optional)
- Section entry keys

You can retrieve this information easily, thanks to the following Gogo shell command:

```
formNavigator:getPossibleConfigurations [optional form ID]
```

This command can be run two ways:

- **With no form ID:** lists every form with all possible categories and entries
- **With a form ID:** lists all categories and entries for the given form

For example, running the Gogo shell command `formNavigator:getPossibleConfigurations sites.form` results in the output shown in the figure below:

```
g! formNavigator:getPossibleConfigurations sites.form
general=categorization,custom-fields,details,documents-and-media,pages,site-template,site-url
social=mentions,ratings
languages=display-settings
advanced=analytics,content-sharing,default-user-associations,maps,recycle-bin
```

Figure 33.7: You can use the formNavigator GoGo shell command to retrieve the available categories and entries for forms.

You can read more about using the GoGo shell in the Felix GoGo Shell Reference Guide.

Now that you have the form ID(s) and available categories and entry keys, you can create a Form Navigator configuration. As mentioned above, Form Navigator configurations can contain an optional context. Form contexts are covered next.

Form Navigator Contexts

A Form Navigator context refers to the context in which a form is being viewed. Form contexts allow you to control dynamically which form sections are displayed. The following contexts are available by default:

`add`: Denotes that the form is being viewed when new content is being created. For example, you could use the `add` context to specify which form sections are visible when someone creates a new site.

`update`: Denotes that the form is being viewed when content is being edited. For example, you could use the `update` context to specify which form sections are visible when someone edits a web content article.

You can create additional contexts by following the steps covered in the Creating Form Navigator Contexts tutorial.

Now that you have the form ID(s), available categories and entry keys, and you understand Form Navigator contexts, you can configure the Form Navigator system settings next.

Configuring Form Navigator System Settings

Open the *Control Panel* and go to *Configuration → System Settings*. Select *Foundation → Form Navigator Configuration*.

Follow these steps to create a new configuration entry:

1. Click the *Add* button to bring up the Form Navigator Configuration Form.

2. Enter the Form Navigator ID you retrieved from the Gogo shell command (`sites.form` for example).

3. Next, enter the Form Navigator entry keys. Enter an optional context (`add` or `update`), category (if there is one), and a comma-separated list of Form Navigator entry keys for each section that you want to display. For example, the value below hides the Documents and Media section under the General tab for the Sites form:

   ```
   general=categorization,custom-fields,details,pages,site-template,site-url
   ```

   The value below provides a configuration for the Sites form when editing a site:

   ```
   update.general=categorization,custom-fields,details,documents-and-media
   ```

   Here is an example configuration for the web content form that has no categories and several entries:

   ```
   content,metadata,permissions,related-assets,schedule,small-image
   ```

   Here is the same configuration with an `update` context:

   ```
   update=content,metadata,permissions,related-assets,schedule,small-image
   ```

   The order of the entry keys specifies the order in which the sections are displayed in the form, from top to bottom. To reorder sections, place the comma-separated list of entry keys in the order you want the sections to appear in the form. You can also move a section from one category to another by adding the entry key to the category in which you want the section to appear.

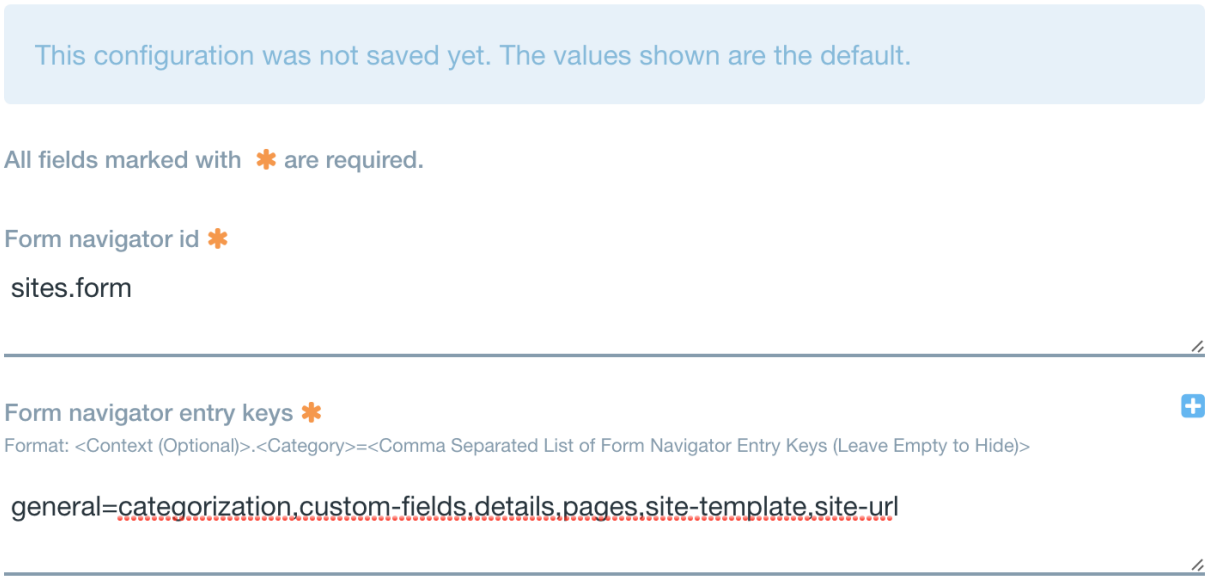


Figure 33.8: You can hide form categories and sections using the Form Navigator Configuration system setting.

```
**Note:** If you want to hide a category, enter the category followed by an
equal sign with no entry keys. For example, `general=` would hide the
General category for the Sites form.
```

---

4. Save your changes.

Now you know how to control the visibility of your Form Navigator form categories and sections!

Related Topics

Custom Fields
    Form Navigator Extensions
    Forms and Validation
    Core JSP Hook

33.6 Custom Fields

Have you ever wondered why there's no *Head Circumference* field in the form for adding users to Liferay DXP? Probably because most sites based on Liferay DXP don't need it. However, as an administrator at the Lunar Resort, you certainly need to know the head circumference of your users so they can be provided with a properly fitting helmet.

Many of Liferay DXP's assets and resources are designed to let you add new fields to their edit forms. Here's the complete list:

- Blogs Entry

- Bookmarks Entry
- Bookmarks Folder
- Calendar Booking
- Document
- Documents Folder
- Message Boards Category
- Message Boards Message
- Organization
- Page
- Role
- Site
- User
- User Group
- Web Content Article
- Web Content Folder
- Wiki Page

Developer Use Case: Adding custom fields to Liferay DXP resources affords flexibility to developers. Suppose you must limit the number of users that can be assigned to a particular role. First an administrator creates a custom field called *max-users* for the role. A developer then creates a module that inserts logic before a user is added to that role. If their logic detects that the maximum number of role users would be exceeded by completing the action, an exception is thrown and the action does not complete. A tutorial is being planned to illustrate how this is accomplished using Custom Fields and a module deployed to @product.

Adding Custom Fields

To add custom fields, find the Custom Fields entry beneath the Control Panel's Configuration heading.

To add a custom field to one of the listed entities,

1. Choose a resource by clicking on it.
2. Click the add ( ) button.
3. Add a Key.

```
**Note:** The Key you enter here is the name of the new field. It's stored
in the database and used by developers to access the custom field with the
`<liferay-ui:custom-attribute />` tag.
```

---

4. Choose the type of field you want to add.

```
**Available Field Types:** There are a number of field types to choose from:

Presets:

- Selection of Integer Values
- Selection of Decimal Values
- Selection of Text Values
- Text Box
- Text Box--Indexed
- Text Field--Secret
- Text Field--Indexed

Primitives

- True/False
- Date
- Decimal number (64-bit)
- Group of Decimal numbers (64-bit)
- Decimal number (32-bit)
- Group of Decimal numbers (32-bit)
- Integer (32-bit)
- Group of Integers (32-bit)
- Integer (64-bit)
- Group of Integers (64-bit)
- Decimal Number or Integer (64-bit)
- Group of Decimal numbers or Integer (64-bit)
- Integer (16-bit)
- Group of Integers (16-bit)
- Text
- Group of Text Values
- Localized Text
```

```
![ At The Lunar Resort, a Head Circumference field is necessary for all users.](./images/custom-fields-user-head-circumference.png)
```

5. Click Save.

That's it.

Once you have a custom field for a resource, go find it in the UI of the resource. First find the UI location for the resource, and all its custom fields are displayed in a Custom Fields panel. For example, consider the Users UI:

Navigate to *Control Panel → Users and Organizations*. Once there click on a User to open the Edit User form. Open the form's Miscellaneous tab and expand the Custom Fields panel. Your custom field is there (its title is whatever you specified in the Key field), ready to be filled out.

Figure 33.9: The Edit User form's Custom Fields panel is found in the Miscellaneous tab.

Editing a Custom Field

You can't change the key or field type of a custom field, but you can delete it and create a new one if necessary. Additionally, once you create the custom field, there are properties you can configure. For example, here's the Head Circumference field's edit page:

Figure 33.10: Custom Fields can be configured after you create them. The exact options depend on the field type you choose.

Edit an individual custom field's permissions by clicking the field's kebab menu (⋮), then *Permissions*. Permission can be granted or removed for these actions:

- Delete
- Permissions
- Update
- View

Custom fields make many of Liferay DXP's entities extensible directly from the administrative user interface. Use them as is, or combine them with some back-end code, and you have yet another powerful, flexible feature at your disposal. As they're fond of saying at The Lunar Resort, "The sky is certainly not the limit."

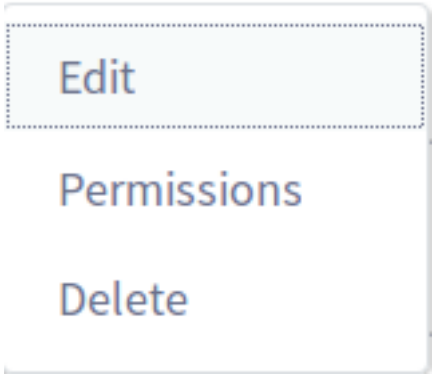


Figure 33.11: You can delete a custom field, edit it, or configure its permissions.

Chapter 34

Targeting Content To Your Audience

Liferay DXP's Audience Targeting app allows you to divide your audience into user segments, target specific content to different user segments, and create campaigns for different user segments. It also allows you to quickly measure the effectiveness of your campaigns. User segments allow you to configure your website to display different assets to different users. Campaigns allow you to display specific content to different user segments for fixed periods of time. They also allow you to measure the interaction of the targeted user segments with the chosen content.

Suppose you'd like to display advertisements to users of your website. You'd like to display one set of advertisements to one user segment and another set of advertisements to another user segment. Maybe this should dynamically change every week based on your marketing goals. Or maybe you have created several sets of news articles for your website that would be interesting to different user segments. You'd like to display news articles to users based on all the user segments to which they belong.

Liferay's Audience Targeting app, available from Liferay Marketplace, allows you to create multiple user segments which are defined by multiple rules based on session attributes, profile attributes, behavior, and information from social networks. Developers can easily create additional rules and rule types with minimal coding efforts. The Audience Targeting App also allows you to create campaigns that target specific user segments. Campaigns last for fixed periods of time and each campaign has a priority. If you are running several campaigns on your website at the same time, the priority field determines which campaign takes precedence. Campaigns allow you to configure different assets to be displayed at different periods of time to the targeted user segment.

The Audience Targeting App adds an *Audience Targeting* section to the Configuration section of the Site Administration area of the Menu and a User Segments Simulator to the Simulation Menu. The following three applications are also included with the Audience Targeting app:

- Campaign Content Display
- User Segment Content Display
- User Segment Content List

You can add these applications to any Liferay DXP page. In the next sections, you'll lean how to use the *Audience Targeting* section of the Site Administration area to manage user segments and campaigns. Then you'll learn how to use each of the Audience Targeting applications and the Audience Targeting Simulator.

Note: The IP Geocoder porlet is an Optional Dependency of Audience Targeting and while it is not required to deploy Audience Targeting, you will need it to access the full functionality of Audience Targeting.

For more information on Optional Dependencies, see our Declaring Optional Import Package Requirements troubleshooting page.

34.1 Managing User Segments

The *Audience Targeting* section of the Configuration section of the Site Administration area of the Menu allows you to manage user segments and campaigns.

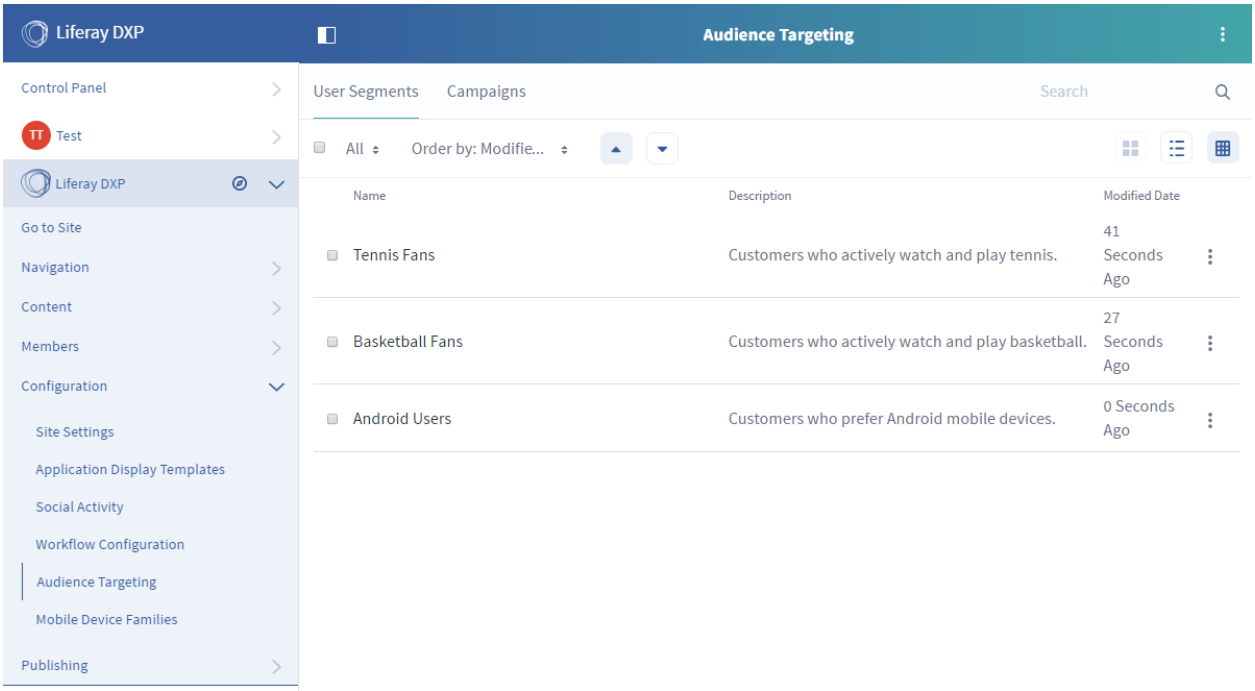


Figure 34.1: Click on *Configuration* → *Audience Targeting* in Site Administration to manage user segments and campaigns for a site.

A user segment represents a subset of the total group of portal users (logged in or not). A user segment is defined by one or more rules that users must match to belong to that user segment. Once the user segment is created, only users who visit the applicable site(s) are added to it. Open the Site Administration menu and click *Configuration* → *Audience Targeting* → *Add User Segment* ( ) to add a new user segment. All the rules that have been deployed appear under the Rules heading. Drag a rule to the right to apply the rule to the user segment. Once a rule has been applied, you can adjust the rule's parameters. For example, once the Gender rule has been applied, you can select *Male* or *Female*. Once the Age rule has been applied, you can select an *Older than* value and a *Younger than* value. For example, you could define a *Women over 30* user segment by applying the Gender rule and selecting *Female*, and applying the Age rule and setting the *Older than* attribute to *30*.

Some rules are also instantiable, meaning you can apply more than one of the same type of rule to a user segment. This allows you to create scenarios where you need to segment your audience based on actions that might occur multiple times (e.g., visiting multiple pages, viewing multiple banners, having several roles, etc.).

Once you've customized the rules for the new user segment, entered a name and, optionally, a description, click *Save* to actually create the user segment.

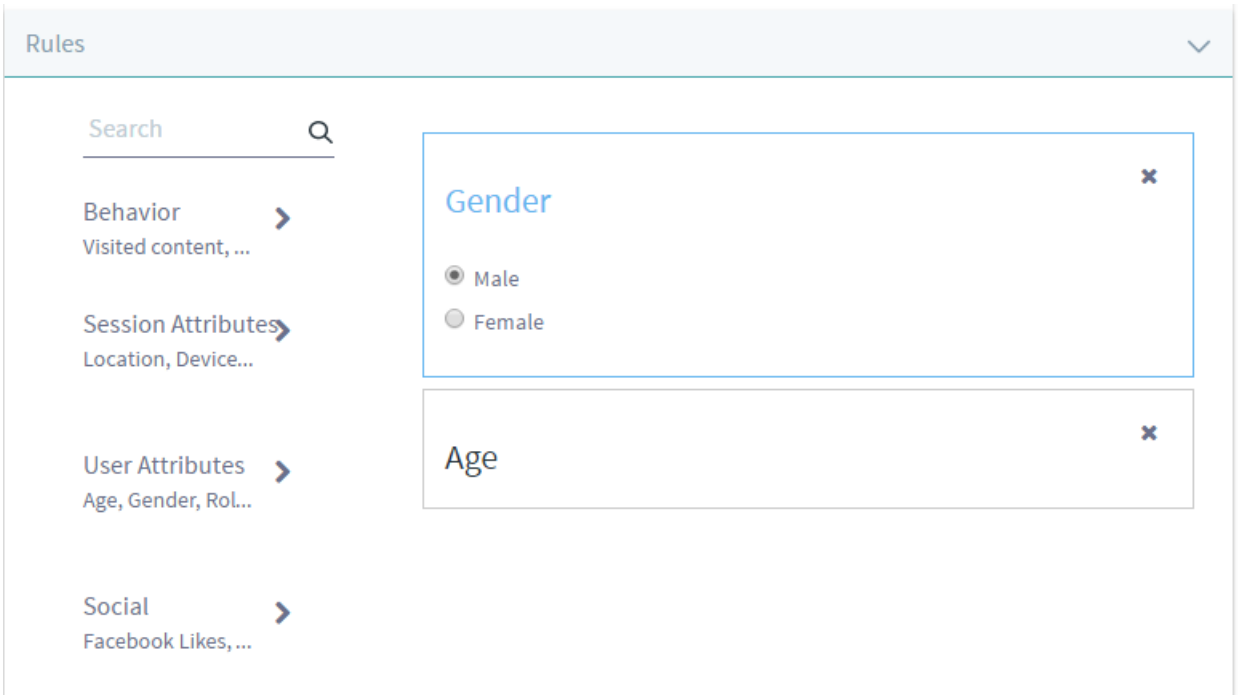


Figure 34.2: After applying the rule, click the rule to expland/collapse it.

Once you've created a user segment, you can open its summary view, which displays relevant data and configurations.

To configure permissions for creating, editing, and deleting user segments, visit the *Users → Roles* section of the Control Panel. Then click on the *Actions → Define Permissions* button corresponding to the role you'd like to configure permissions for, and search for *Audience Targeting*.

Both user segments and campaigns are inherited in your site hierarchy. Therefore, user segments and campaigns defined in the Global scope will be available to any site. User segments and campaigns created in a site will be available to all of its child sites.

User Segment Rules

Custom rules can be created by developers and deployed as OSGi plugins. See the Creating New Audience Targeting Rule Types tutorial for details.

These are some of the rules that are included with the app by default:

- User Attributes
  - Age (from the user profile)
  - Gender (from the user profile)
  - Role (regular role, organization role or site role)
  - Membership (site member, organization member, user group member)
- Session Attributes
  - Location (obtained from the IP address)
  - Browser, Device, Operating system
  - Time

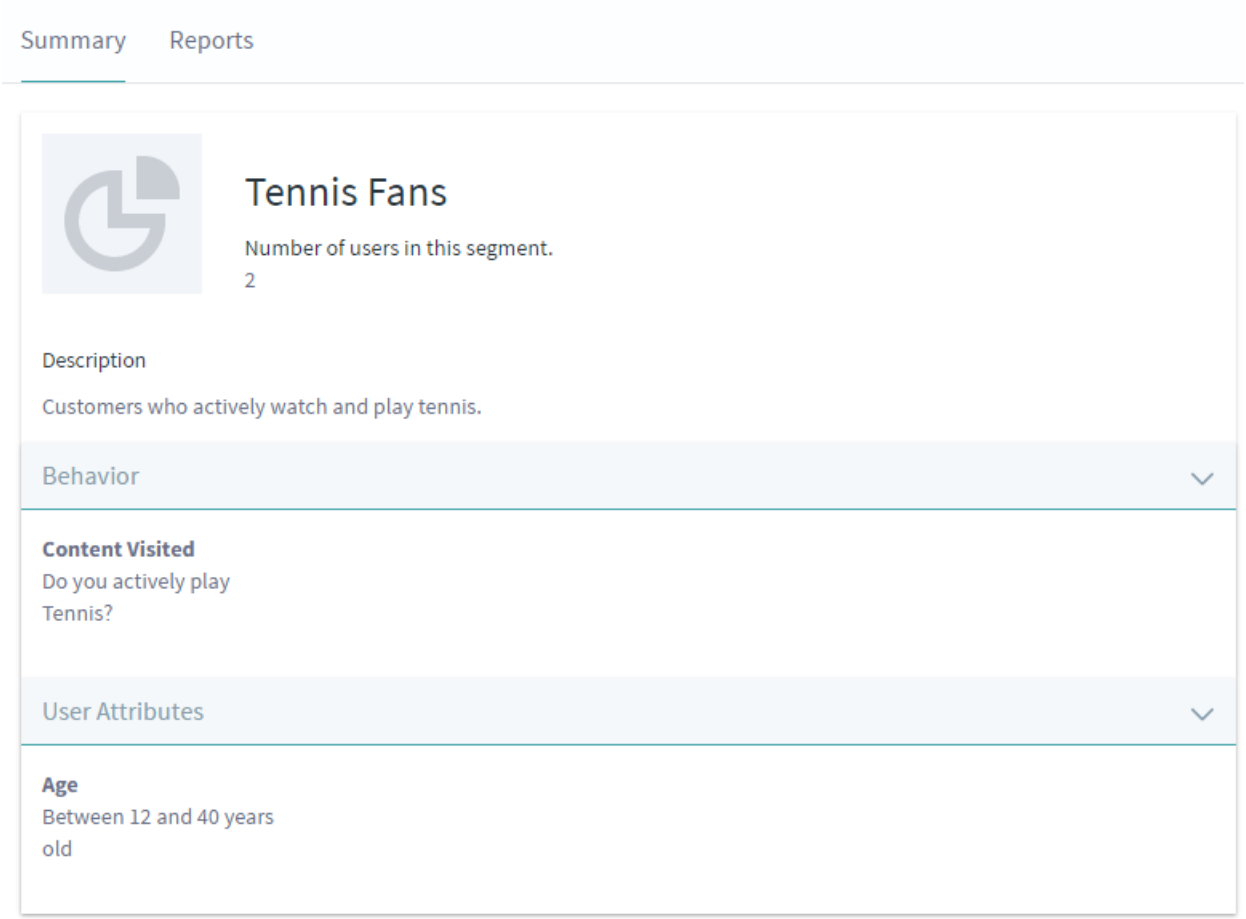


Figure 34.3: Select a pre-existing user segment to view its Summary page.

- Social
  - Like of a specific Facebook page
  - Number of Facebook friends
  - City, Age, Gender, Education, etc. from your Facebook profile
- Behavior
  - Viewed page or content
  - Score Points rule.

The Score Points rule assigns 1 point to a user each time the user views a page or content that's been categorized under the user segment to which the rule has been applied. Once a user exceeds the configured threshold, the user matches this rule. For example, suppose that your website is about sports and you have pages and content about basketball, tennis, and soccer. You would like to divide your audience into three user segments (Basketball fans, Tennis fans, and Soccer fans) in order to display the most relevant content to them on your site's front page. After creating these three user segments using the Score Points rule with a threshold of, say, 20, you should appropriately categorize the content which would be most relevant to each user segment. For example, apply the *Basketball fans* user segment to content about basketball, apply the *Tennis fans* user segment to content about tennis, etc. Now, every time a user (even a guest user) visits a page or views a piece of content categorized for a user segment to which the Score Points rule has been applied,

the user will start accumulating points. Once the user has accumulated enough points, the user will belong to the user segment. After a user has visited more than 20 pages or pieces of content related to basketball, the user will belong to the Basketball fans user segment. Once the user belongs to a user segment, you can use that information to direct more relevant information to the user in your website using the User Segment Content Display application.

Categorizing Pages and Content for User Segments

Each new user segment that's created can be used to categorize pages or content. The Audience Targeting app adds a new *User Segment* select button to the SEO section of pages and Metadata section for assets. These buttons allow you to assign one or more site-scoped or global user segments to the content. This categorization has mainly two purposes:

- Assigning points to users using the Score Points rule
- Showing dynamic lists of content in the User Segment Content List application

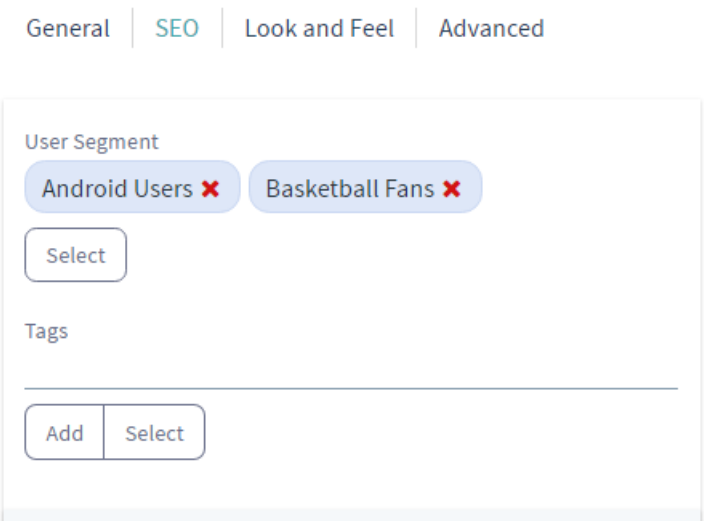


Figure 34.4: Pages and content can be categorized for user segments.

You don't have to create categories for each of your user segments. User segments are distinct from regular vocabularies. The editing screen for both pages and assets contain distinct select buttons for user segments and regular vocabularies.

Another way to display user segments is through the Asset Publisher app. You can enable the Asset Publisher to retrieve assets that have matching categorization with the user segments of the current user. This enhances the Asset Publisher to only display relevant content to the user. To enable *User Segments Filter*, navigate to the Asset Publisher's *Options* (⋮) → *Configuration* menu and select the *Audience Targeting* option. Then enable the *User Segments Filter*.

Next, you'll learn about managing user segment reports.

Managing User Segment Reports

When managing user segments, you can select the user segment name and then select the *Reports* tab to see the list of reports available for each user segment. Click the report name to view the report or *Actions* (⋮) →

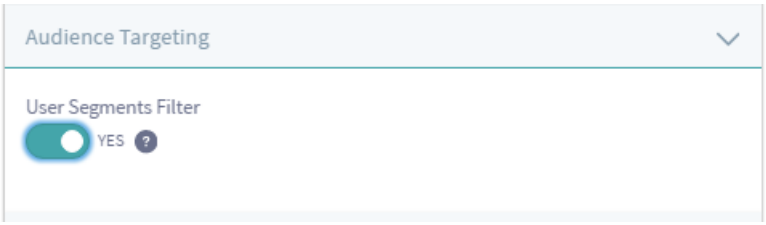


Figure 34.5: Enabling the User Segments Filter retrieves assets that match the current user's user segments.

Update Report to generate a new report. Reports display a summary of interesting information related to each user segment. For example, the Content Views report shows the asset that's been most viewed by users that belong to the user segment.

Title	Type	Event	Count	Last Update
Welcome	Page	View	22	1 Minute Ago
Soccer Apparel	Page	View	12	1 Minute Ago
Basketball Apparel	Page	View	8	1 Second Ago

Figure 34.6: This report displays what pages the user segment has visited.

Reports also display which users belonged to a user segment. This lets an administrator know which users of the site qualified to the particular user segment. Liferay DXP provides a convenient way to export a list of user segment users. Simply click the user report you're interested in and select the *Options* ( ) → *Export* button. This downloads a CSV file with the list of users.

Additional reports can be created by developers and deployed as OSGi plugins. See the Reporting User Behavior with Audience Targeting tutorial for details. Reports are generated daily by default. However, you can generate a report at any time. To generate a new report when currently viewing a report, click on the *Update* button from the *Options* icon ( ) in the top right corner.

Next, you'll discover how to use your user segments in a campaign.

34.2 Managing Campaigns

A campaign represents an effort to expose certain user segments to a set of assets within a specific period of time. To manage campaigns for a site, navigate to *Site Administration* → *Configuration* → *Audience Targeting* → *Campaigns*. To create a new campaign, select the (+) button. You need to select the user segments to target, a start date and an end date, and a priority, as well as a name and, optionally, a description. You also have to indicate whether or not the campaign you create should be active or inactive. When you've entered the required information, click *Save*. The user segments you select when creating a campaign represents the portal users targeted by the campaign. The start and end dates together specify the duration of the campaign. There can be multiple campaigns active at the same time that target the same user segments. In these situations, the priority attribute of the campaigns determines which campaign takes precedence. Finally, you can activate or deactivate a campaign via the *Active* attribute of a campaign. Deactivating a campaign disables the effect of the campaign within the portal. Deactivating a campaign is like deleting the campaign except that a deactivated campaign can be reactivated later. It can be useful to deactivate a campaign if a problem is found with the way content is being displayed. Once the problem has been corrected, the campaign can be reactivated.

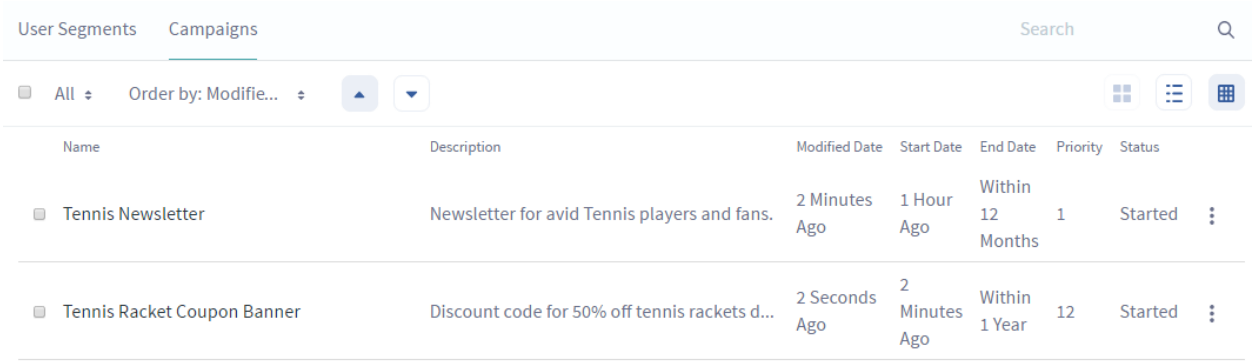


Figure 34.7: Navigate to Site Administration and click *Configuration* → *Audience Targeting* → *Campaigns* to manage campaigns for a site.

For example, suppose you wanted the ability to display certain content (for example, advertisements about your new Android app) to female baseball fans during the months leading up to the World Series. To achieve this, you could use the Gender rule (configured to female), the Device Rule (configured for Android devices), and the Score Points rule to define a user segment called *Female Baseball Fans*. The Score points rule assigns 1 point to a user each time the user visits a page or views an asset categorized under the user segment *Female Baseball Fans*. When a user accumulates a certain number of points (specified by the value of the Score Points rule's *Score Points Threshold* attribute), the user matches this rule. After creating this user segment, you would create a new campaign targeting this segment, select start and end dates, choose a priority, choose *Active*, and then click *Save*. To actually present content to the users belonging to the *Female Baseball Fans* user segment, you need to use the Campaign Content Display application.

Once you've created a campaign, you can open its summary view, which displays relevant data and configurations.

To configure permissions for creating, editing, and deleting user segments, visit the Roles section of the Control Panel. Then click on the *Actions* → *Define Permissions* button corresponding to the role you'd like to configure permissions for, and search for *Audience Targeting*.

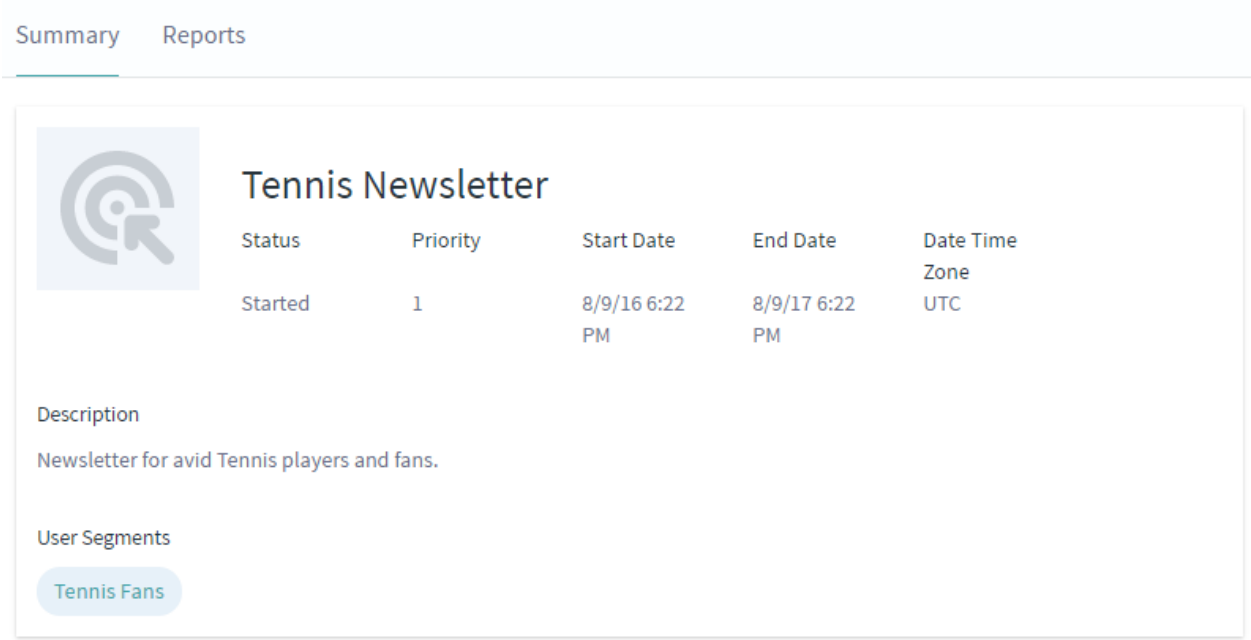


Figure 34.8: Select a pre-existing campaign to view its Summary page.

Managing Campaign Priorities

The priority of your campaigns becomes important when multiple campaigns are running at the same time on your website. The Campaign Content Display application can be configured to display content based on the campaign your users match. When a user matches multiple campaigns, the one with the highest priority takes precedence.

If you have several Campaign Content Display applications around your website configured to display different content per campaign, changing the priority of one campaign automatically affects all the Campaign Content Display applications. Similarly, if a campaign is deactivated or if a campaign's date range is exceeded, all of the Campaign Content Display applications on your website are affected.

Defining Metrics

One of the most interesting features of campaigns is that they allow you to measure the effectiveness of a campaign. This provides your marketing team with real feedback from users. When creating a campaign, you can define the user actions that you want to track. This can be done by defining *Metrics*.

Note: Metrics were previously known as Tracking Actions and were aggregated as part of the campaign editing options. Since 7.0, Tracking Actions have been renamed to Metrics and are aggregated in custom reports. As part of the upgrade process to 7.0, for each campaign containing Tracking Actions, a custom report with the equivalent Metrics is automatically added.

The Audience Targeting app can display reports of how often those actions are triggered. For example, suppose you want to run a campaign for an event that your company is hosting next month. For this event, imagine that you have created a main page for the event which contains a Youtube video and a banner which says *Register Now*. Imagine also that you have a blog entry about the event displayed on several different pages

of your website and a Register page which contains the form to pay for the event. In this campaign, your goal is to get as many people to register as possible. However, you will probably be interested in tracking the following information to see if there is something not working as your team expected:

- Visits to the main page of the event
- Clicks to view the video
- Number of users who watched the video until the end
- Clicks on the Register Now banner
- Views of the blog entry about the event
- Views of the Register form
- Number of users who started to fill out the Register form
- Number of users who completed the registration

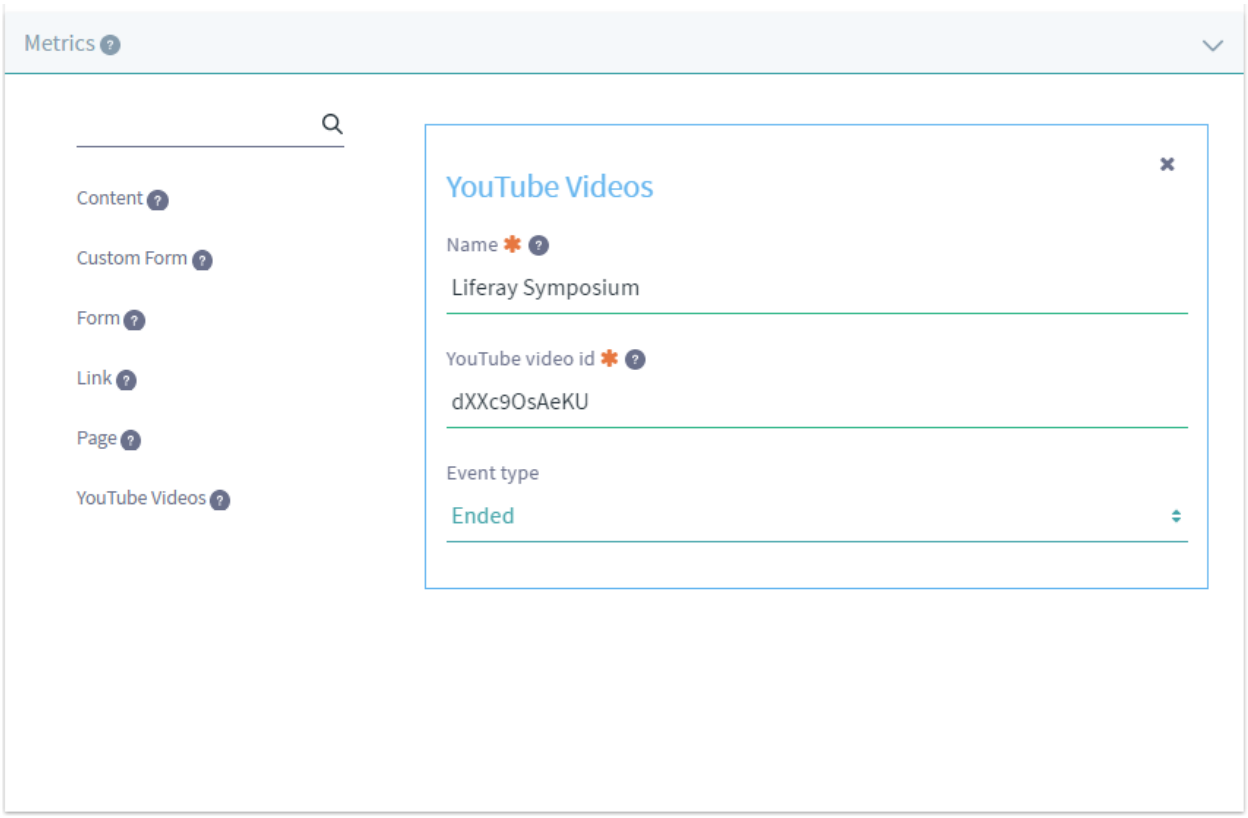


Figure 34.9: Drag and drop metrics to the right to configure a campaign.

You can assign metrics to a campaign report, which is elaborated on in the next section. To access the Metrics palette, select a pre-existing campaign, select the *Reports* tab, and then add a custom report. The Metrics palette is accessible at the bottom of the *New Report* wizard.

You could drag and drop *metrics* from the palette to track all the actions mentioned above. More types of metrics can be created by developers and deployed as OSGi plugins. See the Tracking User Actions with Audience Targeting tutorial for details.

The metrics use an analytics engine called *Audience Targeting Analytics* that can be configured per site or per Liferay DXP installation. You'll learn about this next.

Audience Targeting Analytics To configure the analytics engine per site, go to Site Administration and click *Configuration → Site Settings → Advanced → Audience Targeting Analytics*. To configure it per portal instance, go to *Control Panel → Configuration → Instance Settings → Audience Targeting Analytics*. The following analytics options are available:

- Anonymous Users (not available per site)
- Pages
- Content
- Forms
  - Form Views
  - Form Interactions
  - Form Submits
- Links
- YouTube Videos

Tracking all the actions of all your users (even guest users) can be a heavy load for your server. Therefore, it's best to disable tracking any actions about which you don't need information. For example, Audience Targeting, by default, stores anonymous users' behavior analytics. This stores a large amount of data to the database. If you're not interested in tracking anonymous users, you can turn that functionality off for your Liferay DXP installation by disabling the *Anonymous Users* selector.

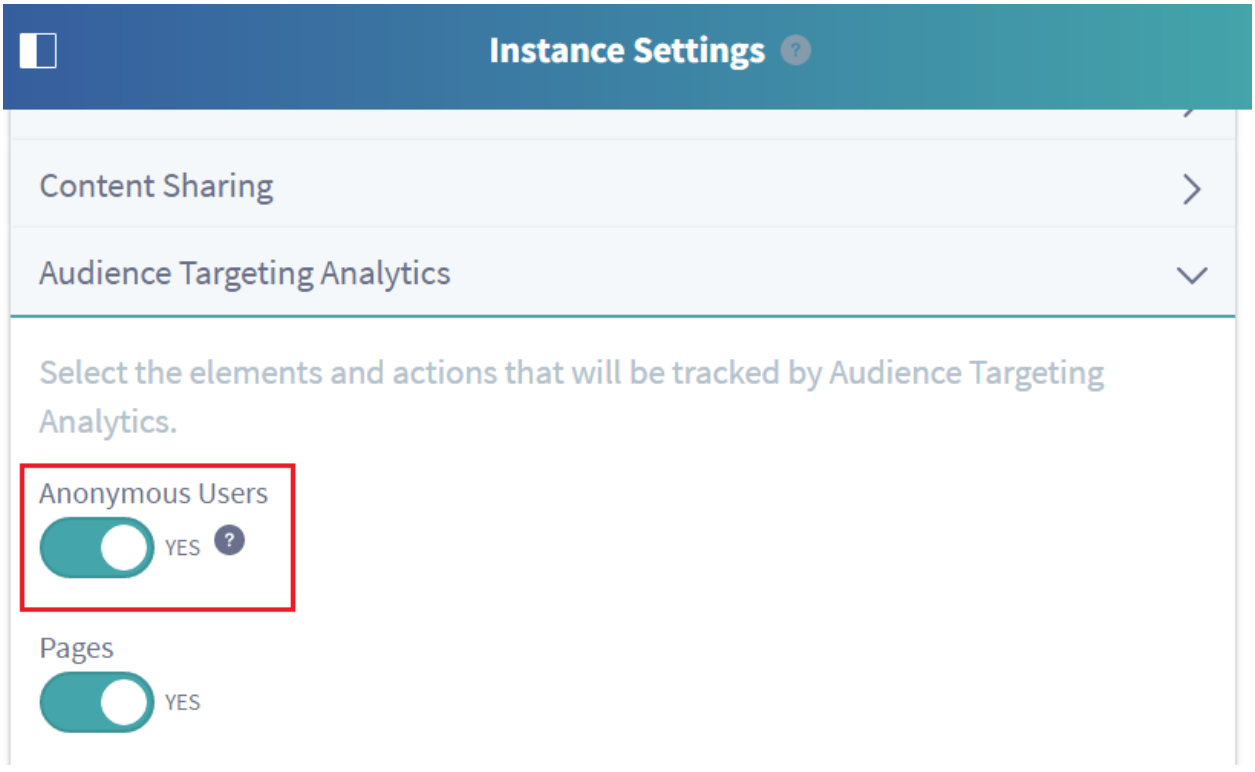


Figure 34.10: There's no need to track anonymous users if you're not interested in their behavior.

Disabling analytics for certain entities means you won't track them using Audience Targeting. Carefully manage analytics to optimize your Audience Targeting experience.

You can also store your analytics data in a separate database schema, which allows for independent scalability. To separate the storage of analytics data from Liferay's database schema, navigate to the Control Panel → *Configuration* → *System Settings* → *Web Experience* and select *Audience Targeting Analytics Storage*. Fill out the external storage fields to point to your alternative database schema.

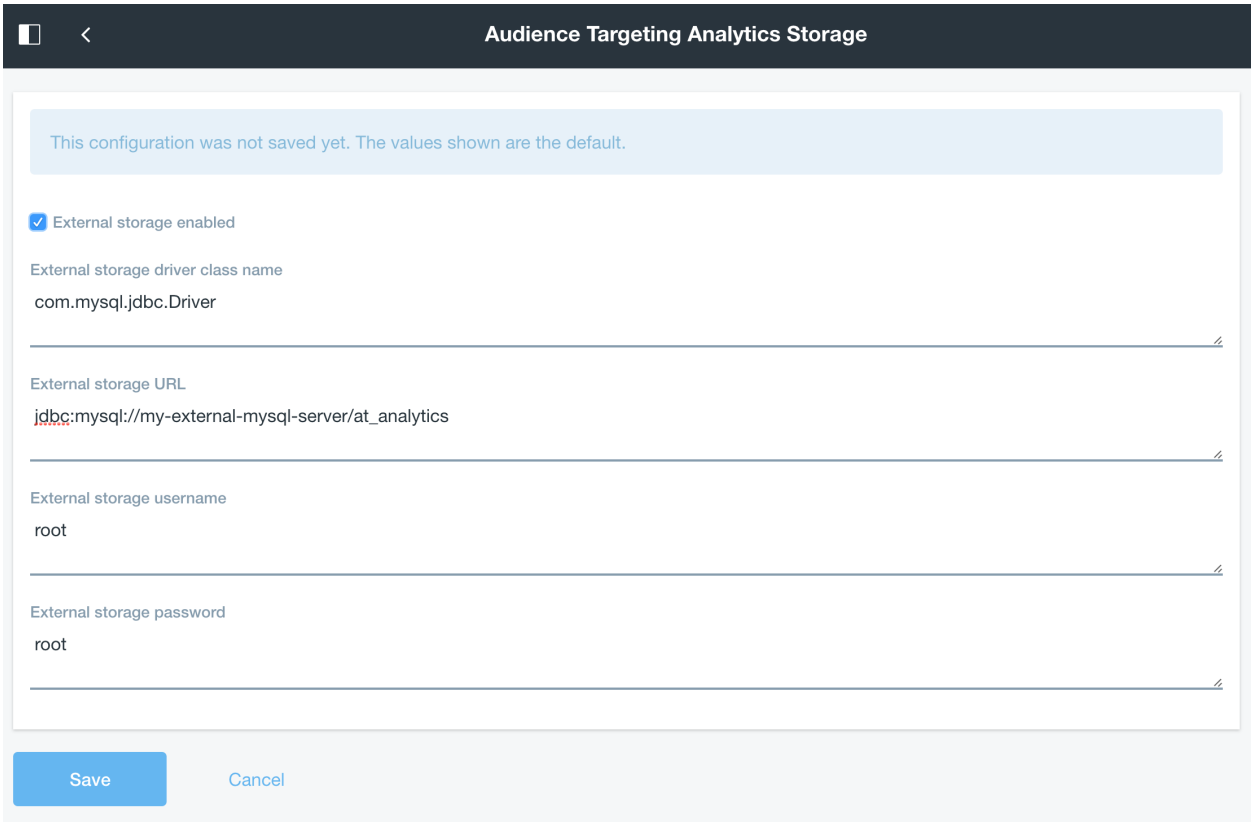


Figure 34.11: By filling out the external storage requirements, you configure your Audience Targeting analytics data to be stored in an alternative database schema.

Once you've saved your external datasource configuration, you must restart the Audience Targeting Analytics component.

1. Navigate to the Control Panel → *Apps* → *App Manager* and select the *Liferay Audience Targeting* app suite.
2. Select the *Options* (⋮) button for the Analytics component and click *Deactivate*.
3. Select the *Options* (⋮) button for the Analytics component again and click *Activate*.

Now your analytics data is stored in an alternative database schema!
Next you'll learn about Campaign Reports.

Campaign Reports

Reports are available for campaigns. You can select the campaign name and click the *Reports* tab to see the list of reports available. More reports can be created by developers and deployed as OSGi plugins. See the Reporting User Behavior with Audience Targeting tutorial for details. You can create a custom report by selecting the a campaign and clicking the *Reports* tab → *Add Custom Report* ( ).

The Content Views report shows the number of times that different assets have been viewed via the Campaign Content Display application by users in the context of the current campaign. For example, if you configured three Campaign Content Display applications around your website to display content for a campaign, the Content View report for the campaign would show how many times that content was interacted with by different users.

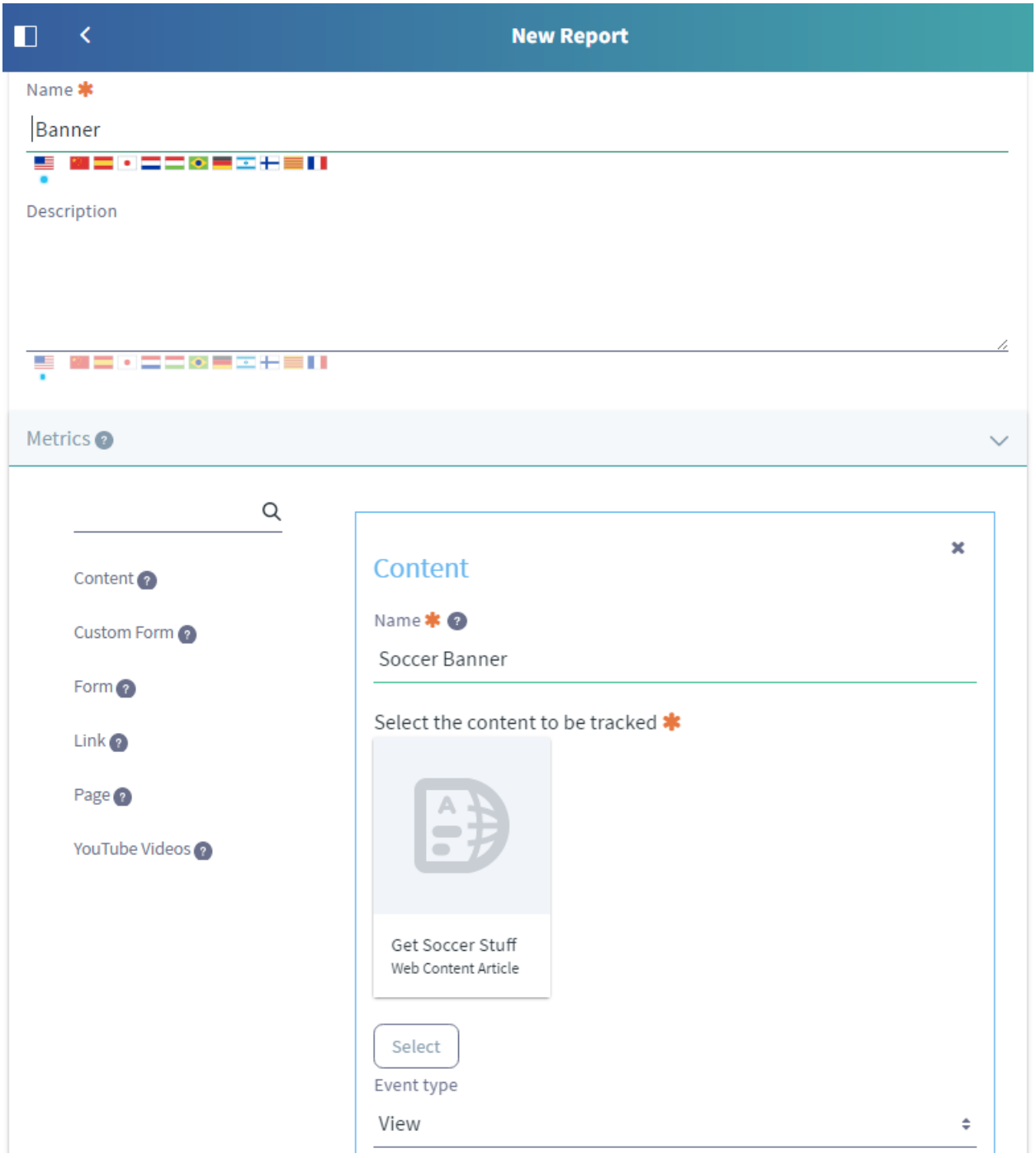


Figure 34.12: You can build your own custom campaign report to fit your needs.

You can track many other user actions by creating a custom report. You can drag and drop different kinds of metrics to track for your campaign, which shows the number of times each metric has been triggered by users. For a complete reference of all report metrics available, see the Audience Targeting Metrics tutorial.

Consider the example that we introduced earlier in the section on metrics: you've created a campaign for an event that your company will host soon. For this event, you have created a main page for the event which contains a Youtube video and a banner which says "Register Now". You also have created a blog post about the event which is displayed on several different pages of your website. Lastly, you have a Register page which contains the form to pay for the event. For this example, a custom Metrics report could show you how many users visited the event page, how many watched the video, how many clicked on the banner, how many viewed the blog post about the event, how many started filling the registration form, etc. This information helps you measure the effectiveness of your campaign. You can use this information to evaluate whether or not the users are following the engagement path you had prepared.

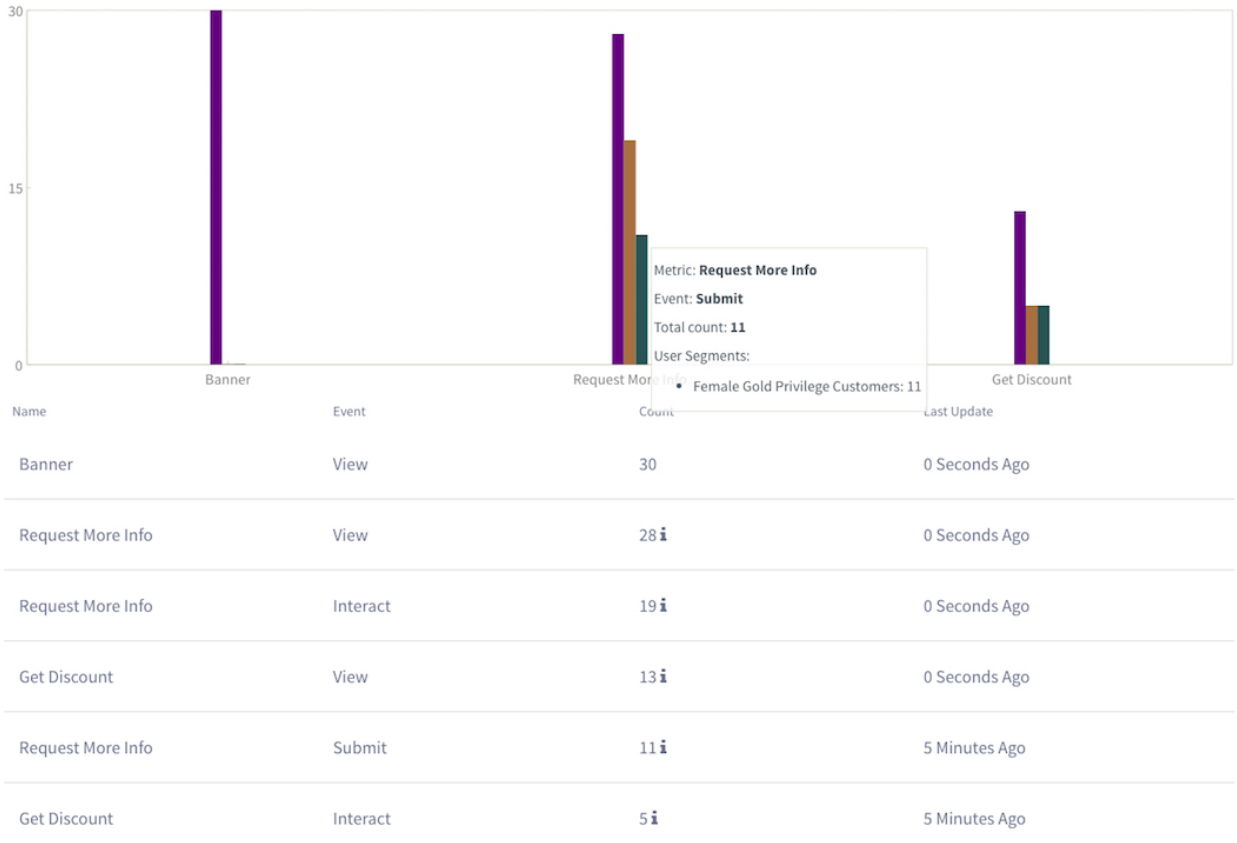


Figure 34.13: This campaign report displays several event types for content in the campaign.

The metrics you apply to a report can be tracked in several different ways. For instance, if you added the *YouTube Videos* metric to your report, you have several different event types to track. Some adminstrators may be interested in how many users played the video, while others are only interested in those that finished the video. You can track these events with the *Event type* field. You even have the option to track *all* events, if your interested in all the metrics for an option.

34.3 Using the Audience Targeting Applications

The Audience Targeting app not only adds the Audience Targeting application to the Site Administration area of the Control Panel, it also includes the following instanceable applications which can be added to any portal page:

- Campaign Content Display
- User Segment Content Display
- User Segment Content List

All of these applications support Application Display Templates (ADTs) so that site administrators can customize the look and feel of the application. Any of Liferay Portal's out-of-the-box Asset Publisher ADTs can actually be re-used for these Audience Targeting applications.

User Segment Content Display

The User Segment Content Display application allows administrators to specify exactly which content to display to a user based on the user segments that the user belongs to. You can specify multiple rules according to the following format:

- If the user [belongs|does not belong] to [any|all] of the following user segments [specify a list of user segments], then display this content: [specify a specific asset].

You can specify any number of *if* clauses when configuring the User Segment Content Display application's rules. However, an *otherwise* clause always follows the last *if* clause so that the application knows what to display if the user doesn't match any user segments. *Don't display anything* is an option for the *otherwise* clause.

For example, you can add a User Segment Content Display application to a page and configure the following rules for it:

- If the user *belongs* to *any* of the following user segments: *Tennis fans*, then display this content: *tennis_picture.jpg*.
- If the user *belongs* to *any* of the following user segments: *Basketball fans*, then display this content: *basketball_picture.jpg*.
- If the user *belongs* to *any* of the following user segments: *Soccer*, then display this content: *soccer_picture.jpg*.
- Otherwise, *Don't display anything*.

Once a User Segment Content Display application has been added to a page and been configured this way, users (even guest users) will see a different image based on the user segment to which they belong. The application won't even be visible to a user if the user doesn't belong to any of the configured user segments.

The User Segment Content Display application allows site administrators to preview the various assets that have been configured to be displayed to different user segments.

Note: You can also provide similar functionality from the Asset Publisher application. You're able to configure a User Segments Filter that will only display content for particular user segments. To learn more about this, see the Categorizing Pages and Content for User Segments tutorial.

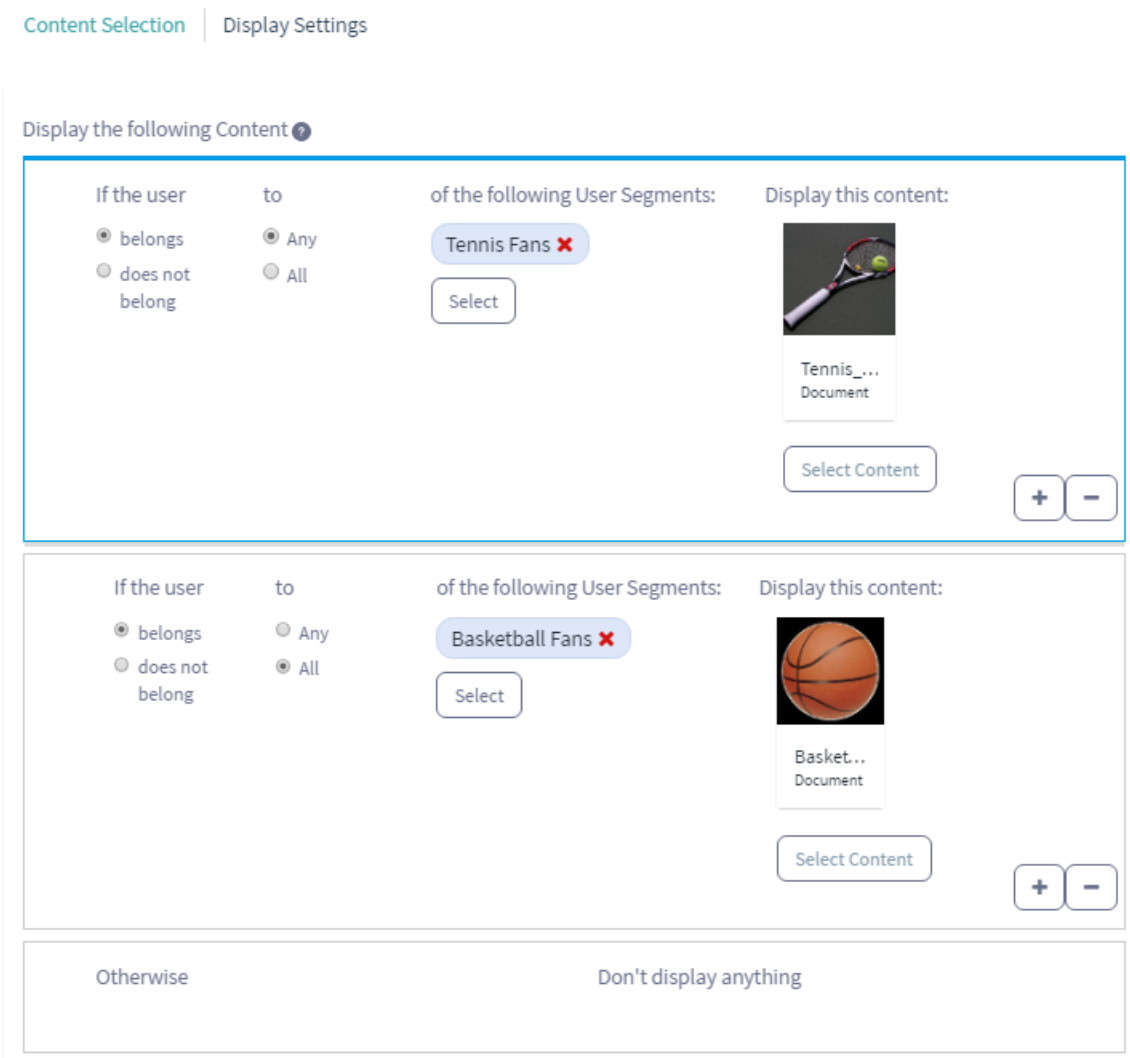


Figure 34.14: You can configure the User Segment Content Display application to display content according to rules that you define in the application's configuration window.

User Segment Content List

The User Segment Content List application displays content that has been categorized for the user segments that match the user segments to which the current user belongs. For example, suppose that your website has several assets categorized under the following user segments: *Tennis fans*, *Soccer fans*, and *Basketball fans*. When a user that belongs to the Tennis fans user segment views this application, the application displays a list of assets categorized for the Tennis fans user segment. If the user belongs to multiple user segments, then a list of articles that have been categorized for any of the matching user segments will be shown.

By default, the User Segment Content List application is configured to display assets of any type that have been categorized for any user segment that matches the current user. However, you can configure the User Segment Content List application to display only assets of specific types. For example, you can configure the User Segment Content List application to only display web content articles and documents. The User Segment Content List application also provides filtering for categories, tags, scopes, etc., offering a wide variety of filtering options so you can display the specific content you want.

Figure 34.15: In the User Segment Content Display application, site administrators can preview the various assets that have been configured to be displayed to different user segments.

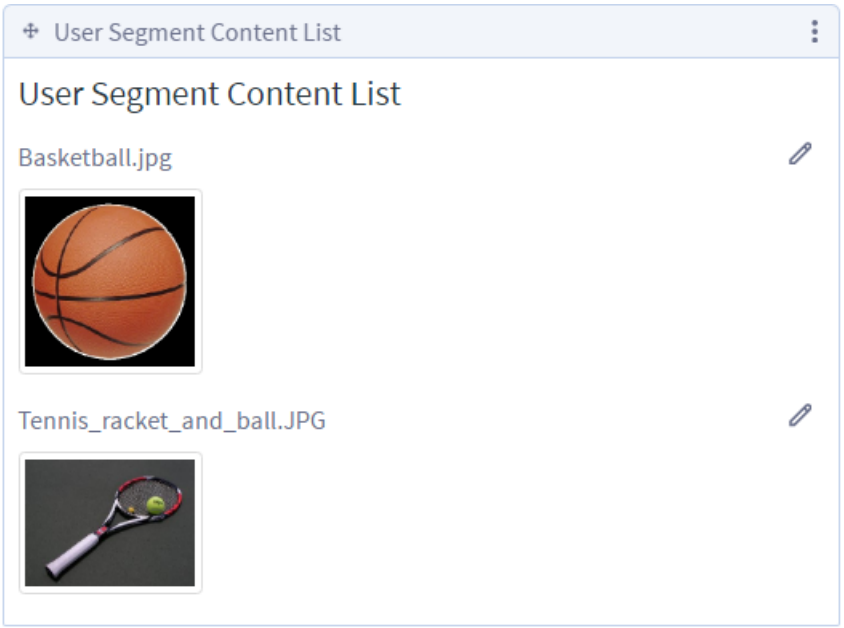


Figure 34.16: The User Segment Content List application displays content that matches the user segments which fit the current user. You can configure the User Segment Content List application to display assets of any kind or only specific kinds of assets.

Campaign Content Display

The Campaign Content Display application is similar to the User Segment Content Display application except that instead of displaying an asset based on the user segments to which a user belongs, it displays an asset based on the campaigns that a user matches. However, the Campaign Content Display application's display rules are simpler than those of the User Segment Content Display application. You can specify multiple rules for the Campaign Content Display application according to the following format:

- If the user belongs to this campaign: [select a campaign], then display this content: [specify a specific asset].

As with the User Segment Content Display application, you can specify any number of *if* clauses when configuring the Campaign Content Display application. An *otherwise* clause always follows the last *if* clause so the Campaign Content Display application knows what to display if the user doesn't match any campaigns. These rules cannot be ordered manually as they can with the User Segment Content Display application. The order of the rules is based on the priority of the campaigns.

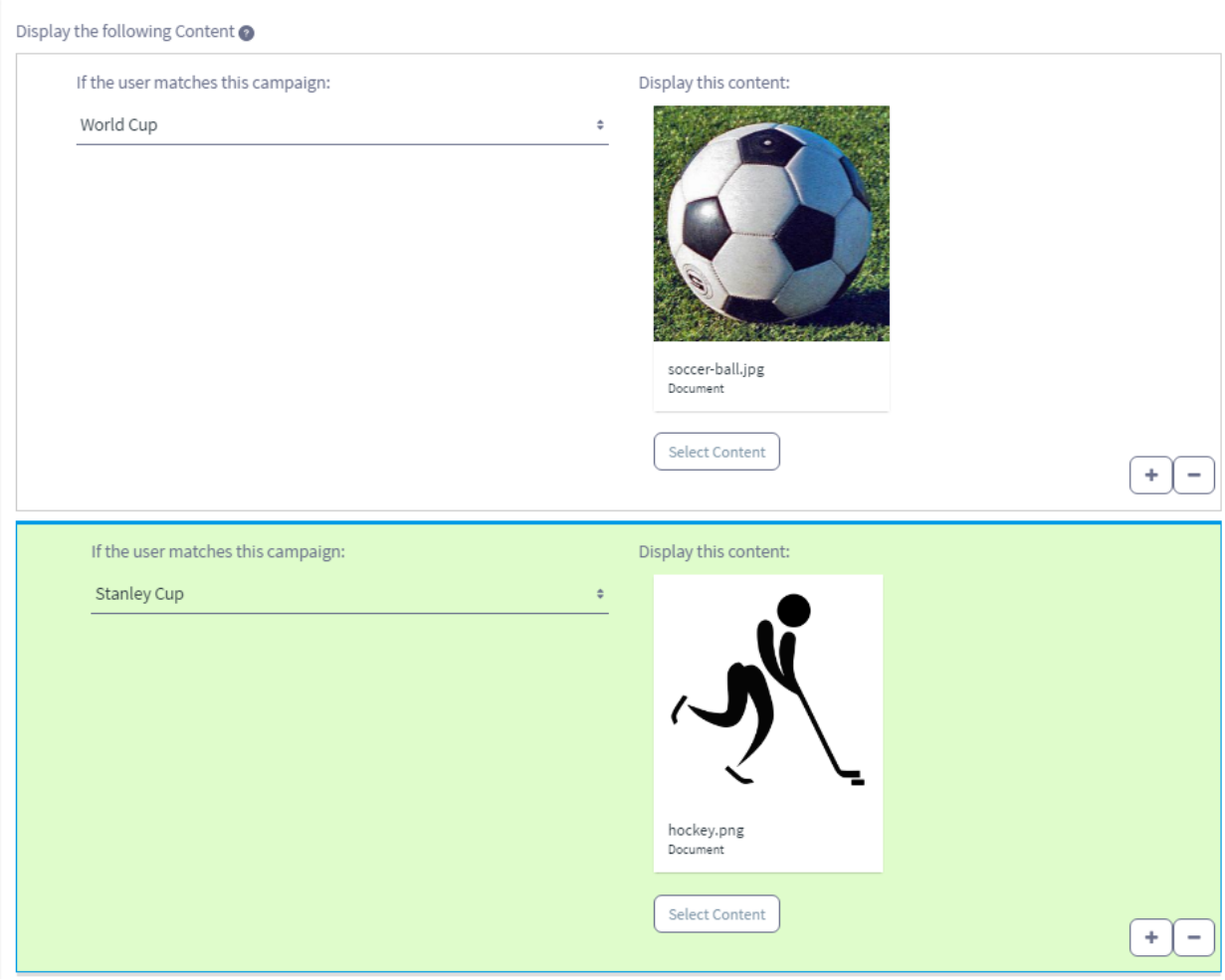


Figure 34.17: The rules for configuring the Campaign Content Display application to display content are similar to the rules of the User Segment Content Display application, but simpler.

For example, suppose that you've created two campaigns called *World Cup* and *Stanley Cup*, designed to target users who are interested in the World Cup of soccer and the Stanley Cup of hockey. You could add a Campaign Content Display application to a page and configure it with the following rules:

- If the user belongs to this campaign: *World Cup*, then display this content: *soccer-ball.jpg*.
- If the user belongs to this campaign: *Stanley Cup*, then display this content: *hockey-.png*.
- Otherwise, display this content: *generic-sports-jersey.jpg*

Once a Campaign Content Display application has been added to a page and been configured this way, portal users who match the World Cup campaign will see a certain image in the application. Users who match the Stanley cup campaign will see a different image in the application. Users who don't match either campaign will see the default image. Of course, once a campaign has ended, no users will match that campaign. Once all campaigns have ended, all users will see the default image.

Note: When Audience Targeting is enabled with a large number of rules, tracking actions, and reports, you may need to adjust some JVM parameters of your server for optimal performance.

The Campaign Content Display application, like the User Segment Content Display application, allows site administrators to preview the different assets that will be displayed for different campaigns.

Asset Publisher

The Asset Publisher is part of Liferay's Web Experience Management suite, but it serves an important function with Audience Targeting. When Audience Targeting is installed, the Asset Publisher gains an additional configuration option: *User Segments Filter*. The User Segments Filter enables the Asset Publisher to display content to users based on their User Segments. In this way, the Asset Publisher provides one of the most important pieces of Audience Targeting: a means of displaying content based on your user segments and campaigns.

When *User Segments Filter* is activated, the only assets that are displayed are those that match the filters configured in the portlet and are categorized under any of the user segments for the current user. Assets not categorized with User Segments don't appear in any Asset Publisher with the *User Segments Filter* enabled.

Next you'll learn how to simulate your user segments and campaigns.

34.4 Simulating User Segments and Campaigns

The Audience Targeting app includes a simulator feature. The Audience Targeting Simulator allows administrators to preview the way portal pages appear to different users. It does so by allowing site administrators to modify the matched user segments. The Audience Targeting Simulator is accessible via the *Simulation* icon () in the top Control Menu. Clicking on the Simulator icon opens the right-side panel which displays a tab related to Audience Targeting: *User Segments*. By default, the Audience Targeting Simulator is only visible to site administrators and users with the View permission for the Audience Targeting Simulator.

Note: Permissions pertaining to the Audience Targeting Simulator and other features of the Audience Targeting app can be granted via the *Actions → Define Permissions* menu of the Roles section in the Control Panel. When defining permissions for a role, go to *Site Administration → Configuration* to find the *Audience Targeting* section.

The User Segments tab of the Audience Targeting simulator displays a list of all the user segments available for the site. You can select or deselect any user segment. This allows site administrators to simulate how the website would appear to different users. Clicking on a user segment makes your website appear to you the way it would appear to that selected user segment. This allows site administrators to navigate around their site as if they were a user that matched the selected user segments.

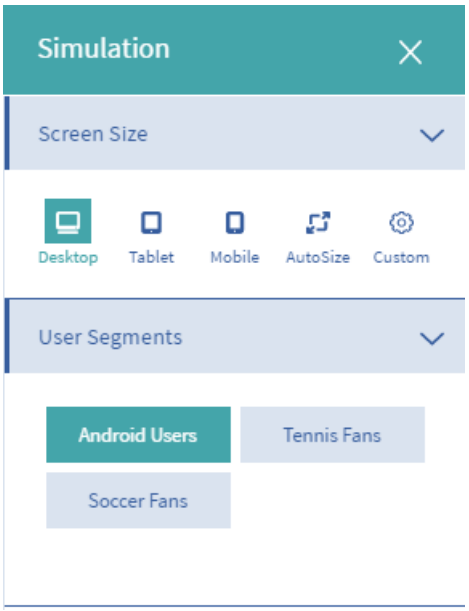


Figure 34.18: Audience Targeting provides a simulator in the Simulation menu.

Next, you'll examine all the Audience Targeting rules available for you to use by default.

34.5 Liferay Audience Targeting Rules

Rules enable Audience Targeting administrators to define how users are classified into the user segments they create.

Basically, a rule evaluates if the current user matches or not one or several conditions. Rules can be configured and combined differently for each user segment to create a very specific classification algorithm. A user must match all rules in a user segment to be classified into it.

Behavior Rules

These rules allow to classify users based on their navigation behavior.

These rules will not work properly if content tracking or page tracking is not enabled. They can be enabled in *Control Panel* → *Configuration* → *Instance Settings* → *Audience Targeting Analytics* (for portal level configuration) or in Site Administration → *Configuration* → *Site Settings* → *Advanced* → *Audience Targeting Analytics* (for site level configuration).

Content Visited Evaluates if the user has visited the selected content.

Page Visited Evaluates if the user has visited the selected page.

Previously Visited Site Evaluates if the user has visited a specific site before visiting the current site. The site's URL can be evaluated based on whether it contains or starts with defined text.

For example, if you'd like to segment users based on whether they navigated to your site from Twitter, you could define a Twitter URL for this rule.

Score Points Evaluates if the user has visited any page or content categorized under this user segment a number of times equals or higher than the configured threshold.

Social Rules

These rules allow you to classify users based on their profiles in social networks (e.g., Facebook).

These rules will not work properly if login through the specific social network (Single Sign On) is not enabled and properly configured. Follow these steps to enable Facebook Single Sign On in Liferay:

1. Sign in to the Facebook Developers site and add a new App. For more information, read the Facebook app developer guide. After going through all the steps, you'll finally have a new app with an Application ID and an Application Secret.
2. Log in as a Portal administrator in your Liferay DXP and go to *Control Panel Configuration → Instance Settings → Authentication → Facebook*. Check *Enabled* and copy your Application ID and Application Secret to their respective fields. Finally, click *Save*.
3. To verify that the Facebook Single Sign On works properly, sign out and then click the Facebook option in the Sign In box. Enter your Facebook username and password in the prompted pop-up and click *Login*. You should be automatically signed in to your Liferay Portal with your Facebook user.

Your Facebook app, by default, only provides access to certain user attributes (e.g., gender, age, number of friends). For the Facebook City, Facebook Education, and Facebook Page Like rules to evaluate properly, you must declare this information accessible in your Facebook app. Liferay DXP cannot read this information from Facebook users without these granted permissions.

Read the Integrating with Facebook section for more information on using Facebook with Liferay DXP.

Facebook Age Evaluates if the age in the user's Facebook profile is in the configured range.

Facebook City Evaluates if the city in the user's Facebook profile matches the configured value.

Facebook Education Evaluates if the education in the user's Facebook profile matches the configured values.

Facebook Friends Evaluates if the user has more or less friends in Facebook than the selected value.

Facebook Gender Evaluates if the gender in the user's Facebook profile matches the selected value.

Facebook Page Like Evaluates if the user likes the configured Facebook page.

User Attributes Rules

These rules allow you to classify users based on the attributes.

These rules will not work properly if the corresponding attributes from the user profile are not available. Follow the given instructions for each rule in such cases.

Age Evaluates if the age in the user's profile is in the configured range.

Custom Field Evaluates if the value set for the selected custom field matches the value from the user's profile.

Gender Evaluates if the gender in the user's profile matches the selected value.

Organization Member Evaluates if the user is member of the selected organization.

Organization Role Evaluates if the user has the selected role assigned in the selected organization.

Regular Role Evaluates if the user has the selected regular role assigned.

Site Member Evaluates if the user is a member of the selected site.

Site Role Evaluates if the user has the selected role assigned in the selected site.

User Group Member Evaluates if the user is member of the selected user group.

User Signed In Evaluates if the user is signed in to the portal.

Session Attributes Rules

These rules allow you to classify users based on their session attributes. Session attributes usually refer to the context in which the user accesses.

These rules will not work properly if the corresponding session attributes are retrieved through any tools that are not currently available or properly configured. Follow the given instructions for each rules in such cases.

Access Time Evaluates if the user is accessing at a time that is within the configured range. Notice that the reference time is that of the server.

Browser Evaluates if the user is accessing with the selected browser.

Device Evaluates if the user is accessing with the selected device.

This rule is based on the existing Device Families. To manage Device Families, go to *Site Administration → Configuration → Mobile Device Families*.

IP Range Evaluates if the user is accessing with an IP with the configured range.

Language Evaluates if the user is accessing with the selected language.

Last Login Date Evaluates if the last login date of the current user was after, before, or between the selected dates.

Location Evaluates if the user is accessing from a country and region that match the configured values.

Operating System Evaluates if the user is accessing with the selected Operating System.

Sign Up Date Evaluates if the Sign Up date of the current user was after, before, or between the selected dates.

In this chapter, you learned how to use Liferay's Audience Targeting app. This app allows you to define custom user segments, target specific content to different user segments, and create campaigns for different user segments. Liferay's Audience Targeting app allows you generate reports so that you can measure the effectiveness of your campaigns. Liferay's Audience Targeting app also includes a simulator so that you can preview how your site would appear to users belonging to different user segments. You can create user segments by applying various rules that reference session attributes, profile attributes, behavior, and possibly other information, such as information from social networks. It's also easy for developers to create additional rules and rule types.

34.6 Audience Targeting Metrics

Metrics enable Audience Targeting administrators to build custom reports and measure the effectiveness of a campaign by tracking certain user actions.

Metrics filter the analytics data gathered by the Audience Targeting Analytics engine to obtain the number of times that a certain action was performed on a given element or content by users and the user segments they belong to.

A custom report may contain multiple metrics of the same type; for instance, metrics for different links. The name field in the metric identifies each value in the report results (e.g., *Link to Marketplace*, *Link to More Info*, etc.).

Metrics can be created by developers and deployed as extensions. Out of the box, Audience Targeting includes metrics to track the most common user actions. These metrics are described below.

Content

Tracks the number of times certain content has been viewed. Use the content selector to set the content to be tracked.

Page

Tracks the number of times a selected page has been viewed. You can track views on both both public or private pages.

Form

Tracks how many users view a form, interact with it (i.e., type or select values in the inputs) or submit it. If you select the *All* option from the *Event type* field, the custom report will show the figures for the three events simultaneously. You must also provide the form you want to track, which is selectable from the *Form* metric.

As a use case, suppose a company wants to organize a webinar to promote one of their products. The marketing team creates a landing page with a subscription form and promotes it on social media. They want to generate a report listing the number of visits to the landing page, form interactions, and form submissions to measure the success of their campaign. This Form metric would allow campaign administrators the ability to track forms in their site.

Link

Tracks how often links are clicked. This helps campaign administrators determine if they're sufficiently visible or helpful.

Similar to forms, you must provide the ID of the link you want to track. If you don't know it, you can inspect the HTML of the page where the link is and extract this information.

YouTube Videos

Tracks how users interact with embedded YouTube videos. You must enter the ID of the YouTube video. You can extract this ID from the video URL as the value for the v parameter. For instance, in the URL `https://www.youtube.com/watch?v=H9xtSO-6aXY` the YouTube video ID is `H9xtSO-6aXY`. Then select one of the available events, or *All* to track all of them. For further reference on the meaning of these events, read the official YouTube API documentation.

Notice that this option only works if the YouTube video is embedded as an iframe. The iframe code is available from the YouTube video's *Share* → *Embed* menu.

34.7 Audience Targeting System Settings

When Audience Targeting is installed on Liferay DXP, there are configuration options provided in System Settings. Audience Targeting configurations made here take effect system-wide. You can find Audience Targeting system settings by navigating to the Control Panel → *Configuration* → *System Settings* → *Web Experience*. The following options are available for the Audience Targeting apps:

- Audience Targeting Analytics Service
- Audience Targeting Analytics Storage
- Audience Targeting Service
- Campaign Content Report
- Campaign Tracking Actions Report
- User Segment Content Report

The settings described in this article are unique to the Audience Targeting application. Some system setting groups may have more configurations than what are defined here; unspecified settings are generic configurations that are applied across many Liferay DXP applications.

Audience Targeting Analytics Service

These settings define the general behavior of the Audience Targeting Analytics modules.

Analytics events check interval: defines the interval (in days) between data check events.

Analytics events max age: defines the max age (in days) of events stored in the analytics tables. When a check event is triggered, entries older than the max age are removed.

Analytics flush interval: defines the interval (in ms) between flushes of the analytics event buffer on the client-side.

Get analytics key: currently only used for testing purposes.

Content tracking enabled: enables Audience Targeting.

Form excluded ids reg exp: defines forms' excluded IDs using regular expressions.

Form tracking enabled: enables tracking capabilities for forms.

Form interaction tracking enabled: enables tracking capabilities for interactions.

Form submit tracking enabled: enables tracking capabilities for form submissions.

Form view tracking enabled: enables tracking capabilities for form views.
Link excluded ids reg exp: defines links' excluded IDs using regular expressions.
Link tracking enabled: enables tracking capabilities for links.
Link click tracking enabled: enables tracking capabilities for clicking events on links.
Page tracking enabled: enables tracking capabilities for pages.
Youtube tracking enabled: enables tracking capabilities for YouTube videos.

Audience Targeting Analytics Storage

These settings configure the analytics engine used by Audience Targeting.

External storage enabled: enables analytics storage.
External storage driver class name: defines the external driver class name for analytics storage.
External storage URL: defines the external storage URL for analytics storage.
External storage username: defines the external storage user name for analytics storage.
External storage password: defines the external storage password for analytics storage.

Audience Targeting Service

These settings define the cleaning task for the record of user assignments (anonymous and registered) to user segments (i.e. the `AnonymousUserUserSegment` table).

Anonymous user user segment check interval: defines the interval (in days) between data check events.

Anonymous user user segment max age: defines the max age (in days) of records stored in the `AnonymousUserUserSegment` table. When a check event is triggered, entries older than the max age are removed.

Campaign Content Report

These settings configure campaign content reports.

Campaign content report check interval: defines the report check interval (in hours) for campaign content. Since the lifetime of events in the Analytics tables is limited to the *Analytics events max age*, reports need to retrieve and process the latest events from Analytics with a shorter time interval.

Campaign Tracking Actions Report

These settings configure campaign tracking action reports.

Campaign tracking action report check interval: defines the report check interval (in hours) for campaign tracking actions. Since the lifetime of events in the Analytics tables is limited to the *Analytics events max age*, reports need to retrieve and process the latest events from Analytics with a shorter time interval.

User Segment Content Report

These settings configure user segment content reports.

User segment content report check interval: defines the report check interval (in hours) for user segment content. Since the lifetime of events in the Analytics tables is limited to the *Analytics events max age*, reports need to retrieve and process the latest events from Analytics with a shorter time interval.

Part II

Deploying Liferay DXP

Chapter 35

Introduction to Liferay DXP Deployment

Liferay DXP is one of the most flexible applications on the market today with respect to database and application server support. Liferay DXP supports a wide variety of databases and application servers, freeing you to use the ones you think are best. Liferay DXP also scales very well. You can install Liferay DXP on a shared hosting account, on a multi-node cluster running a commercial application server, or on anything in between. In fact, Liferay DXP is used successfully in all of these scenarios every day.

35.1 Supported Platforms and Limitations

The list of supported OSes, application servers, databases, and other environments is always available on the Customer Portal at https://www.liferay.com/group/customer under the Support Matrix links.

Applications developed by Liferay, such as Audience Targeting, are available through the Liferay Marketplace. It's possible that some of these applications are supported only on certain environments. This is always specified in the application's description. If you are planning to use these additional products, make sure Liferay DXP is running on a supported environment before installing one of these applications.

35.2 Reference Architecture

When considering deploying Liferay DXP, selecting the right architecture is one of the first decisions you need to make. You need to consider these factors:

- Information Security: Securing sensitive hardware and information from malicious attack and intrusion
- Performance: Supporting the desired number of total users, concurrent transactions, etc.
- Fault Tolerance: Maintaining uptime during unexpected failure or scheduled maintenance
- Flexibility and Scalability: Designing an expandable architecture to support additional features and users without significant redesign

The reference architecture depicted in Figure 1 provides high levels of fault tolerance and flexibility. The architecture contains the following tiers:

- **Firewall:** Intrusion detection and prevention
- **Load Balancer Tier**: Ensures smooth distribution of load between multiple web server resources

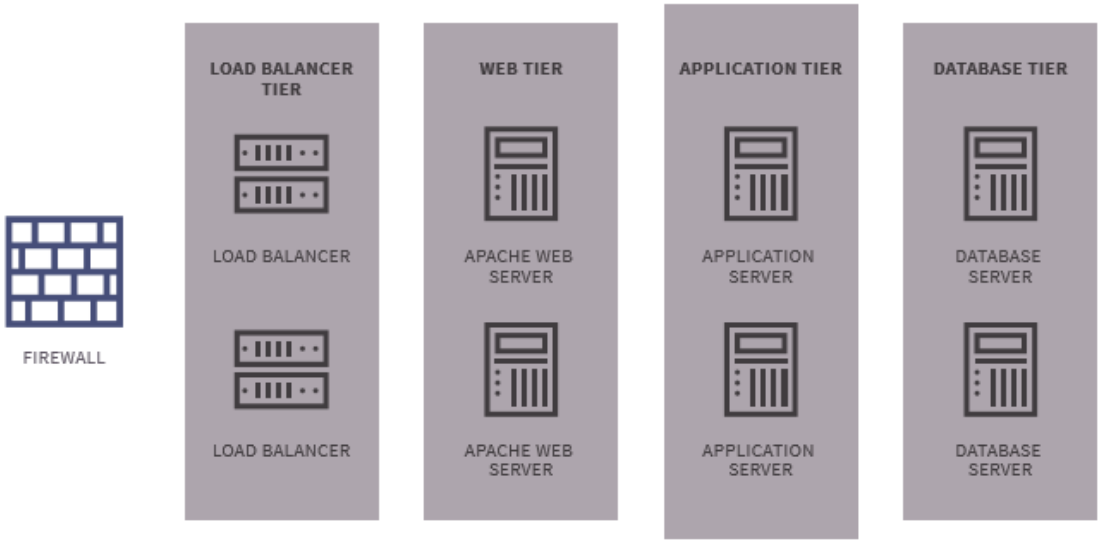


Figure 1 - Digital Enterprise Reference Architecture

Figure 35.1: The 7.0 reference architecture is scalable.

- **Web Server Tier**: Delivers static content like images, rich media, CSS files, etc. Also provides integration modules to single sign-on solutions like CA SiteMinder, Oracle Identity, Ping, etc.
- **Application Tier:** Hosts Liferay DXP supported application servers like Tomcat, JBoss, Oracle Weblogic, and IBM WebSphere (please see 7.0 Compatibility Matrix for additional platforms). Also hosts search engines like Solr and Elasticsearch.
- **Database Tier:** Hosts Liferay DXP supported database servers like MySQL, Oracle, MS SQL Server, IBM DB2, PostgreSQL (please see 7.0 Compatibility Matrix for additional platforms)

The hardware deployed within each tier varies depending on the type of transactions.

Virtualized and Cloud Deployments

While the reference architecture describes a physical deployment, the same concepts may be applied to a cloud based or virtualized deployment. Many Liferay DXP customers choose to deploy on public clouds (e.g. Amazon EC2) or their own private clouds (e.g. VMWare VSX based private cloud). Physical machines can be replaced by appropriate quantities of virtual machines.

In virtualized deployments, it is critical to allocate sufficient CPU resources. For instance, for systems deployed to Amazon AWS, allocated CPUs are calculated using Amazon EC2 Compute Units. However, 1 Compute Unit does not equal to 1 physical CPU or even 1 core on a CPU. In Amazon's terms, each application server equates to roughly a "Cluster Compute Quadruple Extra Large Instance," or 33.5 EC2 Compute Units. Thus, to plan the virtualized / cloud deployment properly, customers must account not only for virtualization overhead, but also for allocation of sufficient CPU resources.

Fault Tolerance

The reference architecture is fault tolerant at every level. With clusters at the web, application, and database tier, you may suffer a catastrophic hardware failure of any node and continue to serve users with little performance degradation.

The depicted reference architecture represents the minimum deployment units to ensure proper fault tolerance within a single data center. You may increase the number of servers within each tier according to your load patterns to achieve a multiplier effect in the number of users the deployment can support while maintaining sufficient fault tolerance.

Multi-data-center fault tolerant architectures are not provided as part of the reference architecture.

Performance

Each deployment's performance characteristics vary depending on the type of activity and the performance of custom application elements. Liferay Engineering has created a series of scenarios to benchmark 7.0's out of the box performance characteristics for content management, collaboration and social enterprise scenarios. Results from these reference architectures have indicated 7.0 can support over 22,000 virtual collaboration users and over 150,000 logins per minute with an average login time of 300 milliseconds. 7.0 accomplished this load within the reference architecture while using no more than 40% of CPU resources in the Web Tier, 86% of CPU resources in the Application Tier, and 50% of CPU resources in the Database Tier.

Scalability

Liferay Engineering's testing has shown 7.0 to scale linearly. Thus, if you know a single application server supports X virtual users and assuming sufficient database and web server resources, you may calculate the total number of application servers required.

Security

The firewall preceding the Load Balancer Tier provides sufficient intrusion detection and prevention. However, depending on your organization's information security requirements, you may introduce additional firewall layers between each tier to further secure the infrastructure.

Environment Types

Before starting to plan your Liferay DXP deployment, it's important to include different environments to be able to maintain the projects. Liferay uses the following terminology for systems:

Development: The environment where the software is written. Most simply it is developer workstations containing integrated development environments with version and dependency handling along with a version control system with working copies on the individual developer workstations.

SIT (System Integration Testing): SIT is where the systems delivered are validated. The is an environment for testing the automation of aggregated components and the dependencies between them.

UAT (User Acceptance Testing): A QA or Test environment. The developers deploy the software here when it's ready for users to verify its functionality.

Production Staging: A pre-production environment that's as close as possible to the production environment for final testing, such as installation, configuration, deployment and load, and performance testing.

Production: The environment that's accessed directly by end users. Because this is the live environment, updates are done using a secure process, particularly if it is a complex clustered environment.

Chapter 36

Deploying Liferay DXP

Liferay DXP is one of the most flexible applications on the market today with respect to database and application server support. Liferay DXP supports a wide variety of databases and application servers, freeing you to use the ones you think are best. Liferay DXP also scales very well. You can install Liferay DXP on a shared hosting account, on a multi-node cluster running a commercial application server, or on anything in between. In fact, Liferay DXP is used successfully in all of these scenarios every day. Some features, such as clustering, are only available with Liferay's Digital Experience Platform.

You'll find that because Liferay DXP is extremely flexible in its deployment options, it is also easy to install. If you already have an application server, you can use your application server's deployment tools to install Liferay DXP. If you don't already have an application server, Liferay provides several application server bundles from which to choose. These are pre-configured application servers with Liferay DXP already installed on them. They're easy to install and with a small amount of configuration can be made into production-ready systems.

Read on to learn how you can prepare to install Liferay DXP.

36.1 Preparing for Install

Installing Liferay DXP is easy. But before you begin, you should answer a few questions.

- Which version of Liferay will you install?
- On which application server will you install Liferay DXP?
- Can you install a Liferay DXP bundle or do you need to install Liferay DXP on an existing application server?
- Which database will you use with Liferay DXP?
- How do you plan to store your data?
- Can your network support Liferay DXP?
- Will you enable Liferay's PACL security feature?

Next, you'll answer these questions and learn the basic steps for installing Liferay DXP.

Obtaining Liferay DXP

Anyone can download Liferay DXP from liferay.com. Click Platform → *Downloads*, and you'll be able to download either the open source version of Liferay DXP or a trial of the commercial version in several

different formats. These include our convenient bundles as well as `.war` files for installing Liferay DXP on your application server of choice.

Liferay enterprise subscribers can download Liferay DXP from the Help Center. You have a choice of the same open source app server bundles as community members, plus a few commercial alternatives, in addition to the `.war` files for manual installation.

So what is a Liferay DXP bundle anyway? A Liferay DXP bundle is an application server with Liferay DXP pre-installed. Using a bundle is the easiest way to install Liferay DXP. Liferay DXP is bundled with a number of application servers; all you need to do is choose the one that best fits your needs. If you don't currently have an application server preference, consider starting with the Tomcat bundle. Tomcat is one of the most lightweight and straightforward bundles to configure. If you have an open source application server preference, choose the server you prefer from the available Liferay DXP bundles. You must have a JDK (Java Development Kit) 8 installed prior to launching Liferay DXP.

Note: Please see the compatibility matrix for information on supported JDKs, databases, and environments.

Please note that Liferay DXP is not able to provide application server bundles for proprietary application servers such as WebLogic or WebSphere, because the licenses for these servers don't allow for redistribution. Liferay DXP's commercial offering, however, runs just as well on these application servers as it does on the others. You'll need to follow our manual installation procedure using a `.war` file to install Liferay DXP on proprietary application servers.

Once you have Liferay DXP, you can plan out your installation. First, determine if you need Liferay DXP Security turned on. Second, prepare your database. Third, install Liferay DXP. Fourth, configure your network. Fifth, configure search. You can install Liferay DXP either by using a bundle or by installing it manually on your existing application server. Next, we'll go over the steps it takes to install Liferay DXP.

Liferay DXP Installation Steps

Before you begin installing Liferay DXP, you should review these basic installation steps:

1. Choose a database server to use with Liferay DXP and create a new database. Determine whether you want Liferay DXP to manage your database connection or your application server to manage your database connection. We recommend that you let Liferay DXP manage your database connection. Liferay DXP can connect with several open source or enterprise level document repositories.

2. Gather mail credentials for sending email notifications to users. Determine whether you want Liferay DXP to manage your mail session or your application server to manage your mail session. Liferay DXP provides a built-in mail session but also supports a JNDI mail session. We recommend that you let Liferay DXP manage your mail session.

3. Install either a Liferay DXP bundle or install Liferay DXP on an existing application server (further instructions below).

4. Choose IPv4 or IPv6. Determine which address format is best for your network (further instructions below).

5. Determine how you'll configure Elasticsearch. Liferay DXP's default embedded configuration is not supported for production use, so you'll have to install Elasticsearch separately, either on the same infrastructure or on its own.

6. Determine whether you'll use Liferay Marketplace or other third party applications. If you will, you should enable Liferay's Plugin Access Control List (PACL) security feature.

7. Configure ports (optional). Liferay's application server (e.g., Tomcat or Wildfly) uses certain ports for purposes like handling incoming HTTP requests, HTTPS requests, or AJP requests, etc. If you start your application server in debug mode, there will be a port listening for a debugger to connect. If desired, you can configure these ports. Please refer to your application server's documentation for information on its default ports and how to configure them.

   Liferay also provides access to its OSGi framework through a configurable port:

   `module.framework.properties.osgi.console=localhost:11311`

   You can override this default property by copying the line above to your `LIFERAY_HOME/portal-ext.properties` file and adjusting the port number.

We'll go through the steps in order, so first we'll look at the Liferay DXP database.

Step 1: Choose a Database Server and Create a New Database

The recommended way of setting up your Liferay DXP database is also the simplest. Liferay DXP takes care of just about everything. You only need to take two simple steps:

1. Create a blank database encoded with the character set UTF-8. Liferay DXP is a multilingual application and needs UTF-8 encoding to display all of its supported character sets.

```
 **Note:** If you plan to migrate from one database vendor to another,
 [configure the database to use the default query result order you expect for entities Liferay DXP lists](/docs/7-0/tutorials/-
/knowledge_base/t/sort-order-changed-with-a-different-database).
```

```
 **Note:** If you use Sybase, configure the database to allow null values by
 default.
```

---

2. Create a database user for accessing this database. Grant this database user all rights, including the rights to create and drop tables, to the blank Liferay DXP database.

```
**Important:** Liferay requires reading from and writing to the database.
The Liferay database user must therefore have permissions to read and write
data.
```

Liferay DXP uses this database user's credentials to connect to the Liferay DXP database either directly or through its application server. During its initial startup, Liferay DXP creates the tables it needs in the database you just created. It does this automatically, complete with indexes.

This is the recommended way to set up Liferay DXP. It enables Liferay DXP to maintain its database automatically during upgrades or when various Liferay DXP plugins that create database tables of their own are installed. This method is by far the best way to set up your Liferay DXP database.

Warning: If you're using an Oracle database, use the `ojdbc8.jar` driver library with at least Oracle 12.2.0.1.0 JDBC 4.2 versioning because data truncation issues have been detected reading data from CLOB columns.

If you choose to set up Liferay DXP's database with the recommended permissions described in this section, you can skip to the next section.

Warning: The instructions below are not ideal for Liferay DXP installations. This procedure is documented here so that enterprises with more restrictive standards can install Liferay DXP with stricter (but sub-optimal) database settings. If it's at all possible, we recommend that you use the method described in the previous section instead of the procedure outlined below.

Even though Liferay DXP can create its database automatically, some enterprises prefer *not* to allow the database user configured in an application server to have the database permissions necessary for Liferay DXP and its plugins to maintain their tables. For these organizations, Select, Insert, Update and Delete are the only allowed permissions. Thus, in this section we explain how to set up the database manually. If your organization *is* willing to grant the Liferay DXP database user the permissions to create and drop tables in the database–and this is the recommended configuration–then simply use the recommended configuration described in the previous section.

1. Create a new, blank, database for Liferay DXP.

2. Grant full rights to do anything to the Liferay DXP database to the Liferay DXP database user.

3. Install Liferay DXP and start it so that it automatically populates its database.

4. Once the database has been populated with the Liferay DXP tables, remove the permissions for creating and dropping tables from the Liferay DXP database user.

There are some caveats to running Liferay DXP like this. Many Liferay DXP plugins create new tables when they're deployed. Additionally, Liferay DXP has an automatic database upgrade function that runs when Liferay DXP is upgraded. If the Liferay DXP database user doesn't have enough rights to create/modify/drop tables in the database, you must grant those rights to the ID before you deploy one of these plugins or start upgrading Liferay DXP. Once the tables are created or the upgrade is complete, you can remove those rights until the next deploy or upgrade. Additionally, your own developers might create plugins that need to create their own tables. These are just like Liferay DXP's plugins that do the same thing, and they cannot be installed if Liferay DXP can't create database tables. If you wish to install these plugins, you will need to grant rights to create tables in the database each time before you attempt to install them.

Liferay DXP offers several configurations to store Documents and Media files by setting the `dl.store.impl=` property. Available options are Simple File System Store, Advanced File System Store, CMIS Store, DBStore, JCRStore, and Amazon S3Store. In addition, Liferay DXP can be connected to various open source and enterprise-level document repositories. All of the repositories are connected to Liferay DXP through hooks available on Liferay Marketplace (see below).

Once you have your database and document repository ready, you can install Liferay DXP on your server.

Step Two: Gather Your Mail Credentials

Liferay DXP uses a mail server to send email notifications. As part of the install, therefore, you will need to have credentials that Liferay DXP can use to connect to your mail server. Specifically, you must have the following information:

- Incoming POP Server and port
- POP User Name
- POP Password
- Outgoing SMTP Server and port
- SMTP User Name
- SMTP Password
- A list of JavaMail properties that would override a default configuration

Once you've gathered this information, you're ready to move on to the next step.

Step Three: Install

The next step is to install Liferay DXP. You can do this in one of two ways: by installing Liferay DXP bundled with an application server, or by installing Liferay DXP manually on an existing application server. Each Liferay DXP installation's Liferay Home contains several folders.

By far the easiest way to get Liferay DXP installed is to use a bundle. Though bundles are pre-packaged for demo purposes, it is very easy to turn them into full, production-ready instances of Liferay DXP.

Step Four: Network Configurations

Liferay DXP supports both IPv4 and IPv6 address formats. By default, Liferay DXP uses IPv4 addresses. If you are using IPv6, you must configure Liferay DXP:

1. In the application server's environment settings, set `-Djava.net.preferIPv4Stack=false`.
2. Create a `portal-ext.properties` file in your portal's Liferay Home directory (if one does not already exist) and set the `tunnel.servlet.hosts.allowed` property to the target hosts you want to allow (e.g., *0:0:0:0:0:0:0:1*).

Step Five: Configure Elasticsearch

Liferay DXP by default ships with an embedded version of Elasticsearch. While this configuration works well for demo purposes, it is not supported in a production installation. After you install Liferay DXP, you must configure it to connect to a standalone Elasticsearch server or cluster. Depending on the size of your installation, this standalone instance of Elasticsearch can reside either on the same machine you have Liferay DXP on or a different machine. For performance purposes, it is better to install it on a separate machine.

Step Six: Liferay Marketplace and Portal Security

The Liferay Marketplace is an integral part of the Liferay DXP experience. The Marketplace plugin is required to be installed on Liferay DXP. The Marketplace plugin enables a host of features that extend beyond just access to the online Liferay Marketplace. Some of the key features the Marketplace plugin enables are

- Liferay Marketplace: direct access to our online Marketplace
- App Manager: ability to install, uninstall, and update apps

- Bundled Apps: easily manage apps that may come bundled with your Liferay DXP
- Developer Apps: ability to manage apps that you're developing
- License Manager: streamlined license management for your Liferay DXP and apps

You will want to make sure Portal Access Control Lists are enabled if you intend to download and install apps from Marketplace. This is an additional layer of security that defines various permissions that apps may or may not have.

The portal installation process attempts to deploy and register the Marketplace plugin automatically. If you're installing Liferay DXP in an environment that would prevent this from happening, you'll have to perform one of several workarounds.

Now that you know where you're headed, you can install Liferay DXP. If you have decided to install Liferay DXP using a bundle, continue with the next section. If you're installing Liferay DXP manually, skip to the section for your application server of choice. Once you have Liferay DXP installed manually or via a bundle, you can move on to configuring search.

36.2 Installing Liferay DXP

Now that you've performed the steps needed to prepare for your installation, you're ready to install Liferay DXP! Since bundles are the easiest way to complete an installation, all the installation steps below assume a bundle has been installed. If you plan to install Liferay DXP manually, please refer to the article for your app server of choice, and then come back here to complete the configuration steps.

Now you're ready. You've created a blank database for Liferay DXP and have gathered the credentials you need for your mail server. The next step is to install Liferay DXP.

Liferay Home

Liferay DXP bundles contain the same folder structure regardless of application server. The top-level folder is named for the release of Liferay DXP. This folder is called *Liferay Home*. This folder is usually the parent folder of the application server's folder. This is why Liferay DXP bundles place the application server inside the bundle's root folder. On a manual installation, the location of this folder varies by application server. In a bundle, it's part of the bundle. If you're doing a manual installation, please refer to the article covering that app server for its location.

In the Liferay Home folder there are folders for various purposes:

- Liferay Home
  - `data`
  - `deploy`
  - `license`
  - `logs`
  - `osgi`
  - `patching-tool` (Liferay Digital Enterprise 7.0 only)
  - [Application Server]
  - `tools`
  - `work`

`data`: This folder is used to store an embedded HSQL database, Liferay DXP's file repository, and Liferay DXP's search indexes. Liferay DXP is initially configured to use the embedded HSQL database but the HSQL

database is primarily intended for demonstration and trial purposes. Portal property `jdbc.default.url` sets the Hypersonic embedded HSQL database location.

`deploy`: To auto-deploy Liferay DXP plugins, copy them to this folder. Legacy style `.war` files, 7.0 style `.jar` files, and `.lpkg` files from Liferay Marketplace are supported. Portal property `auto.deploy.deploy.dir` sets the auto-deploy location.

`license`: Liferay DXP's copyright and version files are here.

`logs`: This folder contains Liferay DXP's log files. The information in Liferay DXP's log files can be quite valuable for system administrators, especially when trying to diagnose a problem. `portal-impl.jar`'s `portal-impl/src/META-INF/portal-log4j.xml` file sets the location for the log files. To override the log file location, you must use an `ext-impl/src/META-INF/portal-log4j-ext.xml` file in an Ext plugin.

`osgi`: All the JAR files and a few configuration files for Liferay DXP's OSGi runtime belong in this folder. Portal property `module.framework.base.dir` sets the OSGi folder location. Here are its subfolders:

- `configs`: Component configuration files go here
- `core`: Liferay DXP's core modules
- `marketplace`: Marketplace applications and application suites
- `modules`: Modules you've deployed
- `portal`: Liferay DXP's non-core modules
- `state`: Contains OSGi internal state files for such things as OSGi bundle installation, bundle storage, and more
- `target-platform`: Target platform index
- `test`: Modules that support test integration
- `war`: WAR plugins you've deployed

`patching-tool`: This folder contains patches for Liferay DXP and files for installing the patches (Digital Enterprise 7.0 only).

`tools`: For portal upgrade and target platform indexer.

`work`: Module Jasper work files.

[Application Server]: The name of this folder is different depending on the bundle you're using. This folder contains the application server in which Liferay DXP has been installed.

If Liferay DXP is unable to create the resources it needs in the Liferay Home folder or if it finds itself running on certain application servers, it creates a folder called `liferay` in the home folder of the operating system user that is running Liferay DXP. In this case, the `liferay` folder becomes Liferay Home. For example, if the operating system user's name was jbloggs, the Liferay Home folder could be `/home/bloggs/liferay` or `C:\Users\jbloggs\liferay`.

Extracting a Liferay DXP Bundle

Getting a Liferay DXP bundle up and running is as easy as uncompressing the archive, possibly copying a JDBC driver, and then starting the application server. Let's use the Liferay DXP Tomcat bundle as an example.

1. Unzip your Liferay DXP bundle.

2. If you're setting up Liferay DXP to be an actual server, copy your database's JDBC driver `.jar` file to `[Tomcat]/lib/ext`. If you're using a supported open source database or if you're setting up Liferay DXP for demo purposes, you can skip this step.

That's it! You've extracted Liferay DXP, and it's ready for use. This is much easier than doing a manual installation on an app server. If, however, that's what you need to do, please at this point click the link on the

left and go through the installation procedure for your app server of choice. When you're finished with the installation (and before you've started Liferay DXP for the first time), come back to this spot, because you need to hook it up to your database.

Connecting Liferay DXP to Your Database

You can connect Liferay DXP through either your app server's data source or the one that ships with Liferay DXP. Because of its suitability for tuning, it is recommended that you use the data source that ships with Liferay DXP. To do this, you'll create a configuration file called `portal-ext.properties`, and you'll place that file in your Liferay Home folder.

The configuration varies by database, of course, so templates for each one are provided in the reference section. To connect your database, therefore, create a text file called `portal-ext.properties` in your Liferay Home folder. Copy the relevant template for your database and paste it into this file.

Now all you have to do is customize it. Enter the proper host name and user and password credentials for your database, and then save the file.

Running Liferay DXP for the First Time

Next, start your app server, or start the Liferay DXP app in your app server. For example, if you're using the Liferay DXP-Tomcat bundle, start Tomcat as if you had downloaded it manually. Tomcat is launched by invoking a script which is found in its `bin` folder. If you open a command prompt or terminal and go to this folder, you can launch Tomcat via the following command on Windows:

```
startup
```

or the following command on Linux/Mac/Unix:

```
./startup.sh
```

The Liferay DXP Tomcat bundle then starts. If you are on Windows, another command prompt window appears with Tomcat's console in it. If you are on Linux, you can see the Tomcat console by issuing the following command:

```
tail -f ../logs/catalina.out
```

Liferay DXP writes log files to folder `[Liferay Home]/logs`.

The first time Liferay DXP starts, it'll take a while to create all of its database tables. Once it has successfully started, it automatically launches a web browser that displays Liferay DXP's Basic Configuration page. If for some reason your browser doesn't load the Basic Configuration page, launch your web browser and navigate to http://localhost:8080.

Using Liferay DXP's Setup Wizard

Liferay DXP's Setup Wizard runs when you start Liferay DXP for the first time. To make it easy to configure Liferay DXP, the first thing you see when browsing to your newly installed Liferay DXP bundle is a setup wizard. The title of the setup wizard page is Basic Configuration. This page provides a convenient way to make an initial Liferay DXP configuration.

There are two sections of the wizard: the portal and the administrator. For the portal, you need to supply the following information:

Portal Name: the name of the portal you're powering with Liferay DXP.

Default Language: choose the default locale of your portal.

Time Zone: select your Liferay DXP instance's default time zone.
For the administrator, you need to supply the following information:
First Name: the first name of the default administrator user
Last Name: the last name of the default administrator user
Email: the email address of the default administrator user

Note: the administrator user's email domain is used as the Liferay DXP instance's default domain (i.e., the `company.default.web.id` portal property).

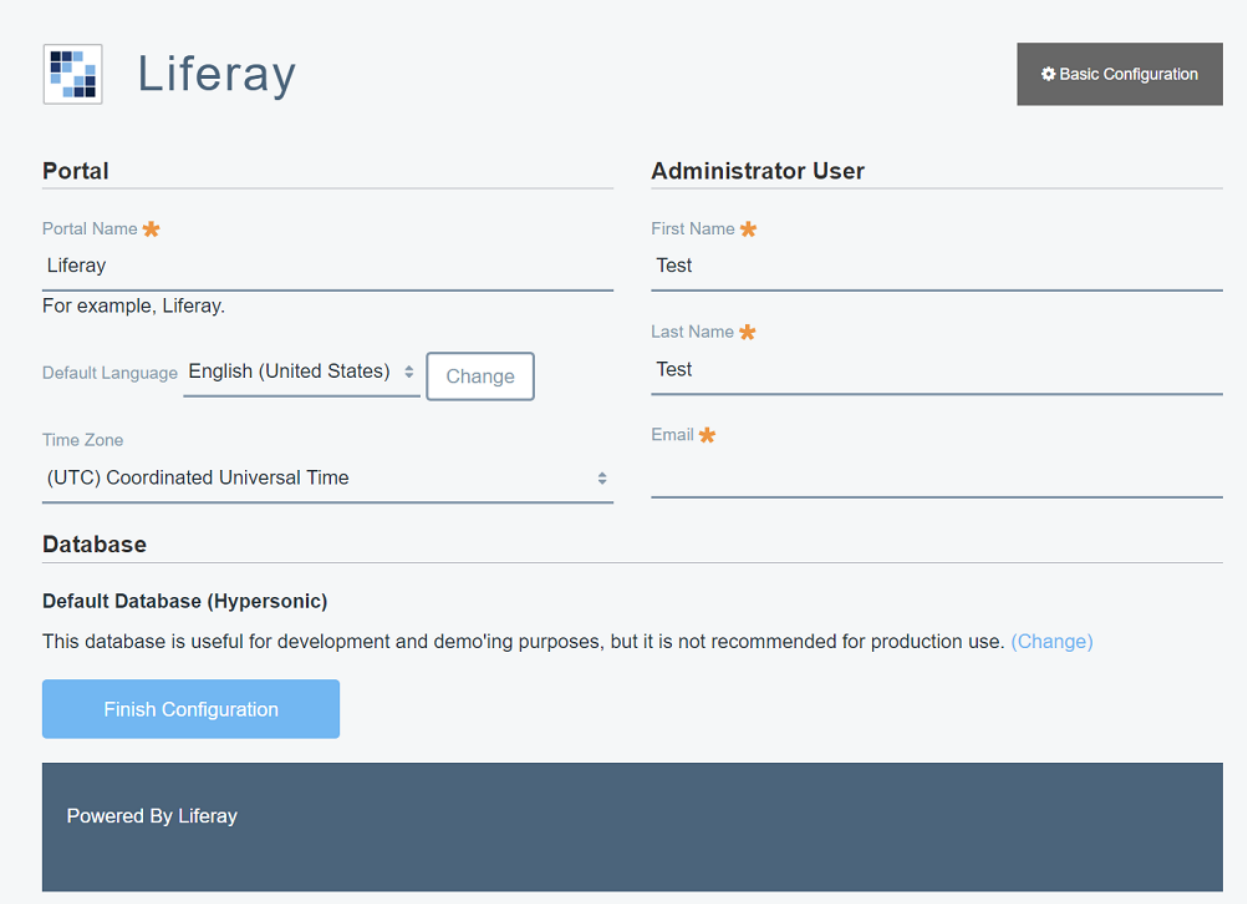


Figure 36.1: Supply the information for your portal and your portal's default administrator user on the Basic Configuration page.

The Basic Configuration page also includes a checkbox labeled *Add Sample Data*. If you check this box, sample data is added to Liferay DXP's database. This data includes users, sites, and organizations. The sample data allows many Liferay DXP features to be showcased. If you're installing Liferay DXP on your own machine to explore its features, the sample data will probably be useful. If, however, you're installing Liferay DXP on a real server, you should start with a clean system.

Once you've filled out the form, click *Finish Configuration*. The setup wizard creates a `portal-setup-wizard.properties` file which stores the settings that you entered. When you begin customizing your portal's configuration, however, you should use the `portal-ext.properties` file you created earlier. All the possible properties that can be placed in this file are documented in our reference documentation.

Tip: The wizard is an extremely helpful tool, especially if you're setting up Liferay DXP for the first time. If you're a Liferay DXP veteran and you already have your various properties set up, you can disable the setup wizard. If you disable the setup wizard, you must configure everything manually from the `portal-ext.properties` file. To disable the setup wizard, enter `setup.wizard.enabled=false` in your `portal-ext.properties` file. Note that property values in `portal-setup-wizard.properties` (the file created in Liferay Home by the setup wizard) override property values in `portal-ext.properties`.

After you've entered the information requested by the Basic Configuration page, Liferay DXP should bring you to its home page. You should set up your mail configuration next.

Configuring Mail

Now that Liferay DXP is up and running, log in as the administrative user you created in the setup wizard. Click the menu icon and then go to Control Panel → Server Administration → Mail, and have your mail credentials ready.

Fill out the form. You're asked for the following information:

Incoming POP Server: The hostname for a server running the Post Office Protocol. Liferay DXP checks this mailbox for incoming messages, such as message board replies.

Incoming Port: The port on which the POP server is listening.

Use a Secure Network Connection: Use an encrypted connection when connecting to the POP server.

User Name: The user ID Liferay DXP should use to log into the POP server.

Password: The password Liferay DXP should use to log into the POP server.

Outgoing SMTP Server: The hostname for a server running the Simple Mail Transfer Protocol. Liferay DXP uses this server to send emails, such as password change emails and other notifications.

Outgoing Port: The port on which the SMTP server is listening.

Use a Secure Network Connection: Use an encrypted connection when connecting to the SMTP server.

User Name: The user ID Liferay DXP should use to log into the SMTP server.

Password: The password Liferay DXP should use to log into the SMTP server.

Manually specify additional JavaMail properties to override the above configuration: If there are additional properties you need to specify, supply them here.

When you're finished setting up your mail configuration, click *Save*.

Your next step for basic Liferay DXP configuration is to convert the search implementation from its default demo mode into a production-ready mode.

36.3 Installing Liferay DXP Manually

The easiest way to install Liferay DXP is to use a Liferay DXP bundle. However, this is not always possible. Some organizations have an existing infrastructure into which Liferay DXP must be installed. Other organizations have standardized on a particular application server. Liferay DXP has been designed to work well with many leading application servers. Even if you have to manually install Liferay DXP on an existing application server, the procedure is straightforward. Before you get started, note that there are two distinct approaches to managing Liferay DXP's data source and mail session. Let's review these options.

Using Data Sources

Liferay DXP provides two ways to configure your data source:

- Use Liferay DXP's built-in data source
- Use your application's server's JNDI data source

Liferay DXP recommends that you use the built-in data source. Liferay DXP's data source is configured by a number of properties that are set in a properties file. By default, you can enter database connection information on the Basic Configuration page that appears when Liferay DXP starts for the first time. The setup wizard stores the information you entered in a configuration file called `portal-setup-wizard.properties` in your Liferay Home folder. Liferay DXP's built-in data source uses this information to connect to the database.

Although Liferay DXP recommends that you use the built-in data source, that's not the only option. Some organizations prefer to use the data source provided by their application server of choice. In this case, a JNDI lookup provides a handle to the data source and the application server manages the connection pools. So you can configure Liferay DXP to use your application server's data source, if that's desired.

To do this, you'll need to create your own configuration file and skip the setup wizard. Since you'd be creating this file *after* the wizard anyway, this isn't such a big deal. We show you how to configure a JNDI data source in the Manual Configuration section below.

Since mail sessions are configured similarly to data sources, we'll look at them next.

Using Mail Sessions

As with databases, you have two ways to configure your mail server:

- Use Liferay DXP's built-in mail session
- Use your application server's mail session

Liferay DXP recommends using the built-in mail session. After you've started Liferay DXP, you can configure a mail server through Liferay DXP's Control Panel. Liferay DXP's default configuration looks for a mail server on the same machine on which Liferay DXP is running and it tries to send mail via SMTP to this server. If this is not your configuration, you'll need to modify Liferay DXP's defaults. To do this, you can use a `portal-ext.properties` file in your Liferay Home folder (see below).

To use your application server's mail session, you must create it in your application server. It should point to your mail server. Once you've created a mail session in your application server, you're ready to point Liferay DXP to it. You can do this through your `portal-ext.properties` file or through Liferay DXP's Control Panel.

You now have all the background information you need to decide whether to use Liferay DXP's built-in data source or the one provided by your application server. Similarly, you can now decide whether to use Liferay DXP's mail session or your application server's mail session. If you're planning to use Liferay DXP to manage both your database connection and mail session, great! When you start Liferay DXP, simply enter your database connection information on the Basic Configuration page and then enter your mail server information through the Control Panel.

If you're planning to let your application server manage your database connection, you can't use Liferay DXP's setup wizard. If you want to configure your application server to manage either your database or mail server, you'll have to follow the instructions in the Manual Configuration section below. The Liferay DXP Installation documentation for each specific application server also includes instructions for configuring your application server to manage Liferay DXP's database connection and mail server.

Manual Configuration

If you want your application server to manage either your database connection or mail server (or both), you'll need to manually create this configuration. Create a text file called `portal-ext.properties` in your Liferay

Home folder. This file overrides default properties that come with Liferay DXP. The first setting you'll override is the default configuration that points Liferay DXP to the embedded HSQL database.

As stated above, there are two ways to set up the connection:

- Use Liferay DXP's built-in data source
- Use your application server's data source

Use the setup wizard if you're using Liferay DXP's data source. If you want to use your application server's pool, continue with this procedure.

If you want to use your application server's data source, you will have to create a connection pool in your application server that points to your database. The connection pool should be called `jdbc/LiferayPool`. You can find instructions for how to do this in the installation documentation for each application server that Liferay DXP supports. To tell Liferay DXP to use your `jdbc/LiferayPool` connection pool, add the following directive to your `portal-ext.properties` file:

```
jdbc.default.jndi.name=jdbc/LiferayPool
```

Next, install Liferay DXP according to the instructions for your application server. Once it's installed, you can set up the mail configuration.

For mail, you should use Liferay DXP's Control Panel to create the configuration. Go to *Control Panel → Server Administration → Mail* and enter your settings for your mail session settings. If, however, you're setting up a lot of Liferay DXP machines and they're all going to have similar mail configurations, it's easier to do the configuration once and then copy the configuration file to multiple machines. In this case, you'll want to use the `portal-ext.properties` file. To use the built-in mail session, use the following properties and customize their values for your environment:

```
mail.session.mail.pop3.host=localhost
mail.session.mail.pop3.password=
mail.session.mail.pop3.port=110
mail.session.mail.pop3.user=
mail.session.mail.smtp.auth=false
mail.session.mail.smtp.host=localhost
mail.session.mail.smtp.password=
mail.session.mail.smtp.port=25
mail.session.mail.smtp.user=
mail.session.mail.store.protocol=pop3
mail.session.mail.transport.protocol=smtp
```

To use your application server's mail session, create it first. Then specify it in the `portal-ext.properties` file:

```
mail.session.jndi.name=mail/MailSession
```

When you've finished, save the file.

All the instructions above assumed that you wanted to install Liferay DXP at the root context of your server. But what if that isn't the case? Next, you'll see how to use a different context for Liferay DXP.

Making Liferay DXP Coexist with Other Java EE Applications

By default, Liferay DXP is configured to sit at the root (i.e., /) of your application server. Dedicating your application server to running only Liferay DXP is a good practice. This allows your portal environment to be separated from your web application environment. This is generally a best practice for portals which, by definition, are application development platforms in and of themselves. For this reason, your Liferay DXP

instance is likely to be hosting many applications and even integrating several of them together on a single page. For this reason, you should design your system so your portal environment has all the resources it needs to do this. Configuring it so it is the sole consumer of any other `.war` files that get deployed to the application server helps to make sure your system performs optimally.

If, however, you want Liferay DXP to share space on an application server with other applications, you can. In this instance, you might not want to make Liferay DXP the default application in the root context of the server. If you want to install Liferay DXP in a context other than the root context, follow the instructions from your app server vendor. No additional steps are necessary.

36.4 Installing Liferay DXP on Tomcat 8

7.0 bundled with Tomcat 8 is available on the Help Center (DXP) or the Community Downloads page (Portal CE). The Tomcat bundle contains JARs, scripts, and configuration files required for deploying Liferay DXP. Copying these files from the Liferay DXP Tomcat bundle facilitates installing Liferay DXP on an existing Tomcat application server.

Whether you copy bundle files (recommended) or download and create the files, you must download these *Additional Files* for DXP or Portal CE:

- Liferay DXP WAR file
- Dependencies ZIP file
- OSGi Dependencies ZIP file

Installing Liferay DXP manually requires these basic steps:

- Installing Liferay DXP dependencies to your application server
- Configuring your application server for Liferay DXP
- Installing the Liferay DXP WAR file to your application server

You'll see the term *Liferay Home* used in this installation guide. *Liferay Home* refers to the folder containing your Tomcat server folder. When Liferay DXP is installed on Tomcat, the Liferay Home folder contains the Tomcat server folder as well as `data`, `deploy`, `license`, and `osgi` folders. You'll also see the terms `$CATALINA_HOME` and `$CATALINA_BASE` used in this guide. `$CATALINA_HOME` refers to the folder that contains Tomcat's binaries, while `$CATALINA_BASE` refers to the folder that contains your configuration–just as you're used to from Tomcat's generic installation. If you download a Liferay Tomcat-bundle, both refer to the same folder (just as with a standard Tomcat installation). This folder is usually named `tomcat-[version]` or `apache-tomcat-[version]`.

Installing Liferay DXP Dependencies

Liferay DXP depends on many JARs that are included in the Liferay DXP Tomcat bundle. Some JARs in the bundle are not strictly required but can still be useful. If you don't have a Liferay DXP Tomcat bundle, you can download the required JARs from third-parties, as described below.

1. If you downloaded a Liferay DXP Tomcat bundle, extract the bundle to a temporary location of your choosing. You'll copy a number of resources from this bundle to your Tomcat server as you manually install Liferay DXP.

2. If you have a Liferay DXP Tomcat bundle, copy all the JARs from your bundle's `$CATALINA_BASE/lib/ext` folder to your application server's `$CATALINA_BASE/lib/ext` folder. If the `$CATALINA_BASE/lib/ext` folder

doesn't exist on your application server, create it. If you don't have a Liferay DXP Tomcat bundle, you'll have to individually download the JARs listed below.

Here's a list of the JARs that you need to copy or download to your `$CATALINA_BASE/lib/ext` folder:

- `activation.jar` - http://www.oracle.com/technetwork/java/jaf11-139815.html
- `ccpp.jar` - http://mvnrepository.com/artifact/javax.ccpp/ccpp/1.0
- `com.liferay.registry.api.jar` - https://repository.liferay.com/nexus/content/repositories/liferay-public-releases/com/liferay/com.liferay.registry.api
- `hsql.jar` - http://hsqldb.org/doc/src/org/hsqldb/jdbc/JDBCDriver.html
- `jms.jar`- http://www.oracle.com/technetwork/java/docs-136352.html
- `jta.jar`- http://www.oracle.com/technetwork/java/javaee/jta/index.html
- `jutf7.jar` - http://sourceforge.net/projects/jutf7
- `mail.jar` - http://www.oracle.com/technetwork/java/index-138643.html
- `mysql.jar` - http://dev.mysql.com/downloads/connector/j
- `persistence.jar`- http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html
- `portal-kernel.jar` - http://mvnrepository.com/artifact/com.liferay.portal/com.liferay.portal.kernel
- `portlet.jar` - http://mvnrepository.com/artifact/javax.portlet/portlet-api
- `postgresql.jar` - https://jdbc.postgresql.org/download.html
- `support-tomcat.jar` - http://repo1.maven.org/maven2/com/liferay/portal/support-tomcat

3. Make sure that Tomcat can access the JDBC driver for your database. The list of JARs above includes `mysql.jar` and `postgresql.jar`. If you're using a database whose JDBC driver is not included in the list above, download the driver and copy it to your `$CATALINA_BASE/lib/ext` folder.

4. Create an `osgi` folder in your Liferay Home folder. Then extract the OSGi ZIP file that you downloaded into the `osgi` folder.

   Liferay DXP requires an OSGi runtime, and the `osgi` folder provides this with many required JAR files and configuration files.

Checkpoint:

1. You should have the following files in the `$CATALINA_BASE/lib/ext` folder:

   - `activation.jar`
   - `ccpp.jar`
   - `com.liferay.registry.api.jar`
   - `hsql.jar`
   - `jms.jar`
   - `jta.jar`
   - `jutf7.jar`
   - `mail.jar`
   - `mysql.jar`
   - `persistence.jar`
   - `portal-kernel.jar`
   - `portlet.jar`
   - `postgresql.jar`

- support-tomcat.jar

2. The osgi folder has the following subfolders:

    - configs
    - core
    - marketplace
    - target-platform
    - test

Tomcat Configuration

Next, you need to configure Tomcat for running Liferay DXP.

1. If you're working with a bundle, copy the setenv.bat and setenv.sh files from your bundle to your $CATALINA_BASE/bin folder. If not, create these files.

    These files set a number of JVM options for Catalina, which is Tomcat's servlet container. Among these options is the location of the Java runtime environment. If this environment is not available on your server globally, you must set its location in this file so Tomcat can run. Do this by pointing the JAVA_HOME environment variable for your OS to the location of the Liferay DXP supported JRE:

    ```
    export JAVA_HOME=/usr/lib/jvm/java-8-jdk
    export PATH=$JAVA_HOME/bin:$PATH
    ```

    Once you've done this, configure Catalina's options to support Liferay DXP:

    ```
    CATALINA_OPTS="$CATALINA_OPTS -Dfile.encoding=UTF-8 -Djava.net.preferIPv4Stack=true  -Dorg.apache.catalina.loader.WebappClassLoader.ENABLE_CLEAR_REF
    Duser.timezone=GMT -Xmx1024m -XX:MaxPermSize=384m"
    ```

    This sets the file encoding to UTF-8, prefers an IPv4 stack over IPv6, prevents Tomcat from working around garbage collection bugs relating to static or final fields (these bugs don't exist in Liferay DXP and working around them causes problems with the logging system), sets the time zone to GMT, and gives the JVM 1GB of RAM.

```
**Important:** For Liferay DXP to work properly, the application server JVM
must use the `GMT` time zone and `UTF-8` file encoding.
```

```
These are initial settings. After installation you should tune your system
for performance. As a result of that process, you may find you want to
change particularly the amount of RAM available to Liferay DXP based on how
your system runs.
```

2. If you're working with a bundle, copy the $CATALINA_BASE/conf/Catalina/localhost/ROOT.xml file from your bundle to the corresponding location in your application server. If not, create this file. The ROOT.xml file creates a web application context for Liferay DXP. ROOT.xml looks like this:

```
<Context path="" crossContext="true">

    <!-- JAAS -->

    <!--<Realm
        className="org.apache.catalina.realm.JAASRealm"
        appName="PortalRealm"
        userClassNames="com.liferay.portal.kernel.security.jaas.PortalPrincipal"
        roleClassNames="com.liferay.portal.kernel.security.jaas.PortalRole"
    />-->

    <!--
    Uncomment the following to disable persistent sessions across reboots.
    -->

    <!--<Manager pathname="" />-->

    <!--
    Uncomment the following to not use sessions. See the property
    "session.disabled" in portal.properties.
    -->

    <!--<Manager className="com.liferay.support.tomcat.session.SessionLessManagerBase" />-->
</Context>
```

Setting `crossContext="true"` allows multiple web applications to use the same class loader. The configuration above includes commented instructions and tags for configuring a JAAS realm, disabling persistent sessions, and disabling sessions entirely.

3. Next, you should make sure that the libraries you copied to `$CATALINA_BASE/lib/ext` are loaded when you start your server. If you're working with a bundle, copy the `$CATALINA_BASE/conf/catalina.properties` file from your bundle to your server. If not, open `$CATALINA_BASE/conf/catalina.properties` and replace the line

```
common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,${catalina.home}/lib,${catalina.home}/lib/*.jar
```

with this one:

```
common.loader="${catalina.base}/lib","${catalina.base}/lib/*.jar","${catalina.home}/lib","${catalina.home}/lib/*.jar","${catalina.base}/lib/ext/glob
```

This allows Catalina to access the JARs that you copied to `$CATALINA_BASE/lib/ext`.

4. If you're working with a bundle, copy the `$CATALINA_BASE/conf/catalina.policy` file from your bundle to your server. If not, just replace the contents of the `$CATALINA_BASE/conf/catalina.policy` file with this:

```
grant {
    permission java.security.AllPermission;
};
```

If you want to enable PACL for Liferay DXP, you have to enable Tomcat's security manager and instruct Catalina to use the `$CATALINA_BASE/conf/catalina.policy` file. See the Enabling PACL section for more information.

5. Next, you should make sure that UTF-8 URI encoding is used consistently. If you're working with a bundle, copy the `$CATALINA_BASE/conf/server.xml` file to your server. If not, you can simply make a few edits to `server.xml`. Edit your `$CATALINA_BASE/conf/server.xml` file and add the attribute `URIEncoding="UTF-8"` wherever you see `redirectPort=8443`, in the definition of your connectors (HTTP and AJP). For example:

   ```
   <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />
   ```

   should become

   ```
   <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" URIEncoding="UTF-8" />
   ```

   And

   ```
   <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
   ```

   should become

   ```
   <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8" />
   ```

6. If you're on Unix, Linux, or Mac OS, navigate to your `$CATALINA_HOME/bin` and `$CATALINA_BASE/bin` folders and run the following command

   ```
   chmod a+x *.sh
   ```

   This command makes the shell scripts in Tomcat's `bin` folder executable.

Checkpoint:
At this point, you've finished configuring the application server's JVM settings.

1. The file encoding, user time-zone, preferred protocol stack have been set in the `setenv.sh` (`setenv.bat` for Windows).
2. The default amount of memory available has been increased.
3. The web application context has been declared in `$CATALINA_BASE/conf/Catalina/localhost/ROOT.xml`.
4. The `common.loader` property which allows Catalina to access the JARs in `$CATALINA_BASE/lib/ext` has been updated in `$CATALINA_BASE/conf/catalina.properties`.
5. All Java permissions have been granted in the `$CATALINA_BASE/conf/catalina.policy` file.
6. UTF-8 encoding has been set in `$CATALINA_BASE/conf/server.xml`.
7. If in a Unix/Linux environment, the `chmod a+x *.sh` command has been run to the shell scripts in Tomcat's `bin` folder executable.

Tomcat Database Configuration

The easiest way to handle your database configuration is to let Liferay DXP manage your data source. If you want to use Liferay DXP's built-in data source, you can skip this section. When you first Liferay DXP, you can enter the required database configuration information on the Basic Configuration page.

If you want Tomcat to manage your data source, use this procedure:

1. Make sure your database server is installed and working. If it's installed on a different machine, make sure it's accessible from your Liferay DXP machine.

2. Add your data source as a resource in the context of your web application specified in `$CATALINA_BASE/conf/Catalina/localhost/ROOT.xml`:

```
<Context...>
    ...
    <Resource
        name="jdbc/LiferayPool"
        auth="Container"
        type="javax.sql.DataSource"
        driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost/lportal?useUnicode=true&amp;characterEncoding=UTF-8"
        username="root"
        password="root"
        maxTotal="100"
        maxIdle="30"
        maxWaitMillis="10000"
    />
</Context>
```

   Note that the above resource definition assumes your database name is *lportal*, that you're using MySQL, and that your MySQL username and password are both *root*. You'll have to update these values with your own database name and credentials.

Your Tomcat managed data source is now configured. Next, let's consider your mail session.

Tomcat Mail Configuration

As with database configuration, the easiest way to handle mail configuration is to let Liferay DXP handle your mail session. If you want to use Liferay DXP's built-in mail session, skip this section and use Liferay DXP's Control Panel to configure a mail server after Liferay DXP has been installed and started.

If you want to manage your mail session with Tomcat, use these instructions:

To create a mail session bound to `mail/MailSession`, edit `$CATALINA_BASE/conf/Catalina/localhost/ROOT.xml` and configure your mail session. Make sure to replace the example mail session values with your own.

```
<Context...>
    <Resource
        name="mail/MailSession"
        auth="Container"
        type="javax.mail.Session"
        mail.pop3.host="pop.gmail.com"
        mail.pop3.port="110"
        mail.smtp.host="smtp.gmail.com"
        mail.smtp.port="465"
        mail.smtp.user="user"
        mail.smtp.password="password"
        mail.smtp.auth="true"
        mail.smtp.starttls.enable="true"
        mail.smtp.socketFactory.class="javax.net.ssl.SSLSocketFactory"
```

```
        mail.imap.host="imap.gmail.com"
        mail.imap.port="993"
        mail.transport.protocol="smtp"
        mail.store.protocol="imap"
    />
</Context>
```

Your mail session is configured. Next, you'll make sure Liferay DXP can access your mail session and database.

Configuring Tomcat-managed Database and Mail Sessions

In this section, you'll specify appropriate properties for connecting to your database and mail session.

1. If you will use *Liferay DXP* to manage your data source, simply follow the instructions on the Basic Configuration page that appears when you first start Liferay DXP.

   If you are using *Tomcat* to manage your data source, add the following line to your `portal-ext.properties` file in your *Liferay Home* folder to specify your data source:

   ```
   jdbc.default.jndi.name=jdbc/LiferayPool
   ```

2. If you will use *Liferay DXP* to manage your mail session, you can configure the mail session once Liferay DXP has started. That is, after starting your portal as described in the *Deploying Liferay DXP* section, go to *Control Panel → Server Administration → Mail* and enter the information required to configure your mail session.

   If you are using *Tomcat* to manage your mail session, add the following configuration to your `portal-ext.properties` file to reference that mail session:

   ```
   mail.session.jndi.name=mail/MailSession
   ```

It's just that easy! Before you deploy Liferay DXP, you should configure Portal Access Control Language (PACL) with Liferay DXP on Tomcat.

Enabling PACL

To enable PACL, you need to enable Tomcat's security manager. In the Tomcat Configuration section above, you already added the required permissions to the Tomcat policy configuration file, `catalina.policy`.

- Edit your `$CATALINA_BASE/bin/setenv.sh` (if on Linux, Unix, or Mac OS) or `setenv.bat` (if on Windows) and enable the security manager by inserting the following code into the `CATALINA_OPTS` variable (inside the quotation marks):

  ```
  -Djava.security.manager -Djava.security.policy=$CATALINA_BASE/conf/catalina.policy
  ```

- Check that your `$CATALINA_BASE/conf/catalina.policy` file specifies the required permissions (you should have already addressed this in the Configuring Tomcat section):

  ```
  grant {
      permission java.security.AllPermission;
  };
  ```

To enable the security manager on Tomcat, the server must be started with the `-security` command line option. Shut down your Tomcat instance and then restart it with the following command:

```
./startup.sh -security
```

Tomcat reports the message `Using Security Manager` to your terminal.
Now you have PACL enabled and configured for your portal.

Deploying Liferay DXP

Now you're ready to deploy Liferay DXP using your Liferay DXP WAR file.

1. If you are manually installing Liferay DXP on a clean Tomcat server, delete the contents of the `$CATALINA_BASE/webapps/ROOT` directory. This removes the default Tomcat home page.

2. Extract the Liferay DXP `.war` file to `$CATALINA_BASE/webapps/ROOT`, so that a `WEB-INF` folder, among other files, is in there.

   Now it's time to launch Liferay DXP on Tomcat!

3. Start Tomcat by navigating to `$CATALINA_HOME/bin` and executing `./startup.sh` or `startup.bat`. Alternatively, you can use `./catalina.sh run` or `catalina.bat run`. Using one of the latter commands makes your terminal or command prompt tail Liferay DXP's log file. This can be useful if you want to see the startup activities Liferay DXP performs or debug deployment.

Congratulations on successfully installing and deploying Liferay DXP on Tomcat!

36.5 Installing Liferay DXP on Wildfly

7.0 bundled with Wildfly is available on the Help Center (DXP) or the Community Downloads page (Portal CE). The Wildfly bundle contains JARs, scripts, and configuration files required for deploying Liferay DXP. Copying these files from the Liferay DXP Wildfly bundle facilitates installing Liferay DXP on an existing Wildfly application server.

Whether you copy bundle files (recommended) or download and create the files, you must download these *Additional Files* for DXP or Portal CE:

- Liferay DXP WAR file
- Dependencies ZIP file
- OSGi Dependencies ZIP file

Installing Liferay DXP manually requires these basic steps:

- Installing Liferay DXP dependencies to your application server
- Configuring your application server for Liferay DXP
- Installing the Liferay DXP WAR file to your application server

Liferay Home is one folder above Wildfly's install location. *Liferay Home* refers to the folder containing your Wildfly server folder. When Liferay DXP is installed on Wildfly, the Liferay Home folder contains the Wildfly server folder as well as `data`, `deploy`, `logs`, and `osgi` folders. You'll also see the term `$WILDFLY_HOME` used in this guide. `$WILDFLY_HOME` refers to your Wildfly server folder. This folder is usually named `wildfly-[version]`.

Installing Liferay DXP Dependencies

Liferay DXP depends on many JARs that are included in the Liferay DXP Wildfly bundle. Some JARs in the bundle are not strictly required but can still be useful. If you don't have a Liferay DXP Wildfly bundle, you can download the required JARs from third-parties, as described below.

1. Create the folder `$WILDFLY_HOME/modules/com/liferay/portal/main`. Unzip the the Liferay DXP Dependencies zip file and copy the `.jar` files to this folder.

2. Download your database driver `.jar` file and copy it into the same folder. For example, for MySQL, download the MySQL Connector/J driver and put its `.jar` file into the `$WILDFLY_HOME/modules/com/liferay/portal/main` folder.

3. Download the remaining required JAR and insert it into the same folder.

    - `com.liferay.registry.api.jar`

    Be sure to remove the version number from the JAR file names or update their names where they're defined (you'll see where the `com.liferay.registry.api.jar` is defined next).

4. Create the file `module.xml` in the `$WILDFLY_HOME/modules/com/liferay/portal/main` folder and insert the following contents:

```
<?xml version="1.0"?>

<module xmlns="urn:jboss:module:1.0" name="com.liferay.portal">
    <resources>
        <resource-root path="com.liferay.registry.api-[version].jar" />
        <resource-root path="mysql-connector-java-[version]-bin.jar" />
        <resource-root path="portal-kernel.jar" />
        <resource-root path="portlet.jar" />
    </resources>
    <dependencies>
        <module name="javax.api" />
        <module name="javax.mail.api" />
        <module name="javax.servlet.api" />
        <module name="javax.servlet.jsp.api" />
        <module name="javax.transaction.api" />
    </dependencies>
</module>
```

    Make sure to replace `[version]` with the correct version of the MySQL JDBC driver. If you are using a different database, replace the MySQL `.jar` with the driver JAR for your database (e.g., HSQL, PostgreSQL, etc.).

5. Create an `osgi` folder in your Liferay Home folder. Then extract the OSGi ZIP file that you downloaded into the `osgi` folder.

    Liferay DXP requires an OSGi runtime, and the `osgi` folder provides this with many required JAR files and configuration files.

Checkpoint:

1. At this point, you should have the following files in the `$WILDFLY_HOME/modules/com/liferay/portal/main` folder:

- com.liferay.registry.api.jar
- portal-kernel.jar
- portlet.jar
- a database jar such as the MySQL Connector.

2. The module.xml has listed all jars in the <resource-root-path> elements.

3. The osgi folder has the following subfolders:

- configs
- core
- marketplace
- target-platform
- test

Great! You have your .jar files ready.

Running Liferay DXP on Wildfly 10.0 in Standalone Mode vs. Domain Mode

Wildfly 10.0 can be launched in either *standalone* mode or *domain* mode. Domain mode allows multiple application server instances to be managed from a single control point. A collection of such application servers is known as a *domain*. For more information on standalone mode vs. domain mode, please refer to the section on this topic in the Wildfly 10 Admin Guide. Liferay DXP fully supports Wildfly 10.0 when it runs in standalone mode but not when it runs in domain mode.

You can run Liferay DXP on Wildfly 10.0 in domain mode, but this method is not fully supported. In particular, Liferay DXP's hot-deploy does not work, since Wildfly 10.0 cannot deploy non-exploded .war files in domain mode. Instead, .war files are in the domain/data/content directory. Deployments are only possible using the command line interface. This prevents many Liferay DXP plugins from working as intended. For example, JSP hooks don't work on Wildfly 10.0 running in domain mode, since Liferay DXP's JSP override mechanism relies on the application server reloading customized JSP files from the exploded plugin .war file location. Other plugins, such as service or action hooks, should still work properly since they don't require Wildfly to access anything (such as JSP files) from an exploded .war file on the file system.

Note: This does not prevent Liferay DXP from running in a clustered environment on multiple Wildfly servers. You can set up a cluster of Liferay DXP instances running on Wildfly 10.0 servers running in standalone mode. Please refer to the chapter of this guide on Configuring Liferay DXP for High Availability for information on setting up a Liferay DXP cluster.

Configuring Wildfly

Now you'll make some adjustments in your configuration to support using Liferay DXP.

You can specify the Wildfly server instance's configuration in the XML file $WILDFLY_HOME/standalone/configuration/standa
You must also make some modifications to your configuration and startup scripts found in the $WILDFLY_HOME/bin/ folder. Lastly, you'll need to make some modifications in your $WILDFLY_HOME/modules/. You'll begin with making changes to standalone.xml.

Make the following modifications to standalone.xml:

1. In the <jsp-config> tag, set the Java VM compatibility for Liferay source and class files. They are compatible with Java 8 by default.

```
<jsp-config development="true" source-vm="1.8" target-vm="1.8" />
```

2. Locate the closing </extensions> tag. Directly beneath that tag, insert the following system properties:

```
<system-properties>
    <property name="org.apache.catalina.connector.URI_ENCODING" value="UTF-8" />
    <property name="org.apache.catalina.connector.USE_BODY_ENCODING_FOR_QUERY_STRING" value="true" />
</system-properties>
```

3. Add a timeout for the deployment scanner by setting `deployment-timeout="360"` as seen in the excerpt below.

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
    <deployment-scanner deployment-timeout="360" path="deployments" relative-to="jboss.server.base.dir" scan-interval="5000" runtime-
failure-causes-rollback="${jboss.deployment.scanner.rollback.on.failure:false}"/>
</subsystem>
```

4. Add the following JAAS security domain to the security subsystem `<security-domains>` defined in element `<subsystem xmlns="urn:jboss:domain:security:1.2">`.

```
<security-domain name="PortalRealm">
    <authentication>
        <login-module code="com.liferay.portal.kernel.security.jaas.PortalLoginModule" flag="required" />
    </authentication>
</security-domain>
```

5. Remove the following tags (if necessary):

    - `<location name="/" handler="welcome-content"/>`
    - `<extension module="org.jboss.as.weld"/>`
    - `<subsystem xmlns="urn:jboss:domain:weld:2.0"/>`
    - `<subsystem xmlns="urn:jboss:domain:weld:3.0"/>`

6. Find the `<jsp-config/>` tag and insert the `development="true"` attribute into the tag. Once finished, the tag should look like the following:

```
<jsp-config development="true" />
```

Checkpoint:
Before continuing, verify the following properties have been set in the `standalone.xml` file:

1. A new `<system-property>` has been created.
2. The `<deployment-timeout>` has been set to `360`.
3. A new `<security-domain>` has been created.
4. Four tags have been removed.
5. `<jsp-config development>` has been set to `true`.

Now it's time for some changes to your configuration and startup scripts.

Make the following modifications to your standalone domain's configuration script file `standalone.conf` (`standalone.conf.bat` on Windows) found in your `$WILDFLY_HOME/bin/` folder.

These modifications change the following options: - Set the file encoding - Set the user time-zone - Set the preferred protocol stack - Increase the default amount of memory available.

Make the following edits as applicable to your operating system:

On Windows, comment out the initial `JAVA_OPTS` assignment as demonstrated in the following line:

```
rem set "JAVA_OPTS=-Xms64M -Xmx512M -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m"
```

Then add the following `JAVA_OPTS` assignment one line above the `:JAVA_OPTS_SET` line found at end of the file:

```
set "JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8 -Djava.net.preferIPv4Stack=true -Dsecmgr -Djava.security.policy=$WILDFLY_HOME/bin/server.policy -
Dwildfly.home.dir=$WILDFLY_HOME -Duser.timezone=GMT -Xmx1024m -XX:MaxMetaspaceSize=384m -XX:MetaspaceSize=200m"
```

On Unix, merge the following values into your settings for `JAVA_OPTS`, replacing any matching attributes with the ones found in the assignment below:

```
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8 -Djava.net.preferIPv4Stack=true -Dsecmgr -Djava.security.policy=$WILDFLY_HOME/bin/server.policy -
Dwildfly.home.dir=$WILDFLY_HOME -Duser.timezone=GMT -Xmx1024m -XX:MaxMetaspaceSize=384m -XX:MetaspaceSize=200m
```

Important: For Liferay DXP to work properly, the application server JVM must use the `GMT` time zone and `UTF-8` file encoding.

Make sure you replace the `$WILDFLY_HOME` references with the appropriate directory. You'll notice some Java security options. You'll finish configuring the Java security options in the *Security Configuration* section.

Note: If you plan on using the IBM JDK with your Wildfly server, you'll need to complete some additional steps. First, navigate to the `$WILDFLY_HOME/modules/com/liferay/portal/main/module.xml` file and insert the following dependency within the `<dependencies>` element:

```
<module name="ibm.jdk" />
```

Then navigate to the `$WILDFLY_HOME/modules/system/layers/base/sun/jdk/main/module.xml` file and insert the following path names inside the `<paths>...</paths>` element:

```
<path name="com/sun/crypto" />
<path name="com/sun/crypto/provider" />
<path name="com/sun/image/codec/jpeg" />
<path name="com/sun/org/apache/xml/internal/resolver" />
<path name="com/sun/org/apache/xml/internal/resolver/tools" />
```

The added paths resolve issues with portal deployment exceptions and image uploading problems on a Liferay DXP instance running on Wildfly 10.0.x.

Checkpoint:

At this point, you'll have finished configuring the application server's JVM settings.

1. The file encoding, user time-zone, preferred protocol stack have been set in the `JAVA_OPTS` in the `standalone.conf.bat` file.

2. The default amount of memory available has been increased.

3. If using IBM's JDK, the sun crypto properties have been set in the $WILDFLY_HOME/modules/system/layers/base/sun/jdk/m file.

The prescribed script modifications are now complete for your Liferay DXP installation on Wildfly. Next you'll configure mail and the database.

Database Configuration

If you want Wildfly to manage your data source, follow the instructions in this section. If you want to use the built-in Liferay DXP data source, you can skip this section.

Modify standalone.xml and add your data source and driver in the <datasources> element of your data sources subsystem.

1. First, add your data source inside the <datasources> element.

```
 <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" jta="true" use-java-context="true" use-
ccm="true">
     <connection-url>jdbc:mysql://localhost/lportal</connection-url>
     <driver>mysql</driver>
     <security>
         <user-name>root</user-name>
         <password>root</password>
     </security>
 </datasource>
```

   Be sure to replace the database name (i.e. lportal), user name, and password with the appropriate values.

```
**Note:** If you'd like to change your datasource `jndi-name` to something
different, you'll need to also edit the `datasource` element in the
`<default-bindings>` tag.
```

---

2. Add your driver to the <drivers> element also found within the <datasources> element.

```
 <drivers>
     <driver name="mysql" module="com.liferay.portal"/>
 </drivers>
```

Your final data sources subsystem should look like this:

```
<subsystem xmlns="urn:jboss:domain:datasources:1.0">
    <datasources>
        <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" jta="true" use-java-
context="true" use-ccm="true">
            <connection-url>jdbc:mysql://localhost/lportal</connection-url>
            <driver>mysql</driver>
            <security>
                <user-name>root</user-name>
                <password>root</password>
            </security>
        </datasource>
```

```
            <drivers>
                <driver name="mysql" module="com.liferay.portal"/>
            </drivers>
        </datasources>
    </subsystem>
```

Now that you've configured your data source, the mail session is next.

Mail Configuration

If you want Wildfly to manage your mail session, use the following instructions. If you want to use the built-in Liferay DXP mail session, you can skip this section.

Specify your mail subsystem in `standalone.xml` as in the following example:

```
<subsystem xmlns="urn:jboss:domain:mail:2.0">
    <mail-session jndi-name="java:jboss/mail/MailSession" name="mail-smtp">
        <smtp-server ssl="true" outbound-socket-binding-ref="mail-smtp" username="USERNAME" password="PASSWORD"/>
   </mail-session>
</subsystem>
...
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.binding.port-offset:0}">
...
<outbound-socket-binding name="mail-smtp">
        <remote-destination host="smtp.gmail.com" port="465"/>
    </outbound-socket-binding>
</socket-binding-group>
```

You've got mail! Next, you'll make sure Liferay DXP can connect using your new mail session and database.

Configuring data sources and mail sessions

Now that your data source and mail session are set up, you need to ensure Liferay DXP can access them.

1. First, navigate to the Liferay Home folder, which is one folder above Wildfly's install location (i.e. `$WILDFLY_HOME/..`).

2. If you're using *Wildfly* to manage your data source, add the following configuration to your `portal-ext.properties` file in your *Liferay Home* to refer to your data source:

   ```
   jdbc.default.jndi.name=java:jboss/datasources/ExampleDS
   ```

   If you're using *Liferay DXP* to manage your data source, follow the instructions for using the setup wizard.

3. If you're using *Liferay DXP* to manage your mail session, this configuration is done in Liferay DXP. That is, after starting your portal as described in the *Deploy Liferay DXP* section, go to *Control Panel* → *Server Administration* → *Mail* and enter the settings for your mail session.

   If you're using Wildfly to manage your mail session, add the following configuration to your `portal-ext.properties` file to reference that mail session:

   ```
   mail.session.jndi.name=java:jboss/mail/MailSession
   ```

Before you deploy Liferay DXP on your Wildfly app server, you should enable and configure Java security so you can use Liferay DXP's plugin security manager with your downloaded Liferay DXP applications.

Security Configuration

When you're ready to begin using other people's apps from Marketplace, you'll want to protect your Liferay DXP instance and your Wildfly server from security threats. To do so, you can enable Java Security on your Wildfly server and specify a security policy to grant your Liferay DXP instance access to your server.

Remember, you set the `-Dsecmgr` and `-Djava.security.policy` Java options in the `standalone.conf.bat` file earlier in the *Configuring Wildfly* section. The `-Dsecmgr` Java option enables security on Wildfly. Likewise, the `-Djava.security.policy` Java option lists the permissions for your server's Java security policy. If you have not set these options, you'll need to do so before using Java security.

This configuration opens up all permissions. You can tune the permissions in your policy later. Create the `$WILDFLY_HOME/bin/server.policy` file and add the following contents:

```
grant {
    permission java.security.AllPermission;
};
```

For extensive information on Java SE Security Architecture, see its specification documents at http://docs.oracle.com/javase/7/docs/technotes/guides/security/spec/security-spec.doc.html. Also, see the Plugin Security and PACL tutorial to learn how to configure Liferay DXP plugin access to resources.

Deploy Liferay DXP

1. If the folder `$WILDFLY_HOME/standalone/deployments/ROOT.war` already exists in your Wildfly installation, delete all of its subfolders and files. Otherwise, create a new folder `$WILDFLY_HOME/standalone/deployments/ROOT.wa`

2. Unzip the Liferay DXP `.war` file into the `ROOT.war` folder.

3. To trigger deployment of `ROOT.war`, create an empty file named `ROOT.war.dodeploy` in your `$WILDFLY_HOME/standalone/deployments/` folder. On startup, Wildfly detects the presence of this file and deploys it as a web application.

4. Start the Wildfly application server by navigating to `$WILDFLY_HOME/bin` and running `standalone.bat` or `standalone.sh`.

You're now an expert when it comes to deploying Liferay DXP on Wildfly!

36.6 Installing Liferay DXP on JBoss EAP 6.4

Installing Liferay DXP manually requires these basic steps:

- Installing Liferay DXP dependencies to your application server
- Configuring your application server for Liferay DXP
- Installing the Liferay DXP WAR file to your application server

In this article, you'll step through these basic steps and install Liferay DXP on your existing JBoss EAP 6.4 application server. Before proceeding, you must download these *Additional Files*:

- Liferay DXP WAR file
- Dependencies ZIP file
- OSGi Dependencies ZIP file

Liferay Home is one folder above JBoss's install location. *Liferay Home* refers to the folder containing your JBoss server folder. When Liferay DXP is installed on JBoss, the Liferay Home folder contains the JBoss server folder as well as `data`, `deploy`, `logs`, and `osgi` folders. You'll also see the term `$JBOSS_HOME` used in this guide. `$JBOSS_HOME` refers to your JBoss server folder. This folder is usually named `jboss-eap-[version]`.

Installing Liferay DXP Dependencies

Liferay DXP depends on many JARs that are included in the Liferay DXP JBoss bundle. Some JARs in the bundle are not strictly required but can still be useful. You can download the required JARs from third-parties, as described below.

1. Create the folder `$JBOSS_HOME/modules/com/liferay/portal/main`. Unzip the the Liferay DXP Dependencies zip file and copy the `.jar` files to this folder.

2. Download your database driver `.jar` file and copy it into the same folder. For example, for MySQL, download the MySQL Connector/J driver. and put its `.jar` file into the `$JBOSS_HOME/modules/com/liferay/portal/main` folder.

3. Download the com.liferay.registry.api.jar JAR and insert it into the same folder.

4. Create the file `module.xml` in the `$JBOSS_HOME/modules/com/liferay/portal/main` folder and insert the following contents:

```
<?xml version="1.0"?>

<module name="com.liferay.portal" xmlns="urn:jboss:module:1.0">
    <resources>
        <resource-root path="com.liferay.registry.api.jar" />
        <resource-root path="mysql-connector-java-[version]-bin.jar" />
        <resource-root path="portal-kernel.jar" />
        <resource-root path="portlet.jar" />
    </resources>
    <dependencies>
        <module name="javax.api" />
        <module name="javax.mail.api" />
        <module name="javax.servlet.api" />
        <module name="javax.servlet.jsp.api" />
        <module name="javax.transaction.api" />
    </dependencies>
</module>
```

   Make sure to replace `[version]` with the correct version of the MySQL JDBC driver. If you are using a different database, replace the MySQL `.jar` with the driver JAR for your database (e.g., HSQL, PostgreSQL, etc.).

5. Create an `osgi` folder in your Liferay Home folder. Then extract the OSGi ZIP file that you downloaded into the `osgi` folder.

   Liferay DXP requires an OSGi runtime, and the `osgi` folder provides this with many required JAR files and configuration files.

Checkpoint: 1. Inside the `$JBOSS_HOME/modules/com/liferay/portal/main` folder, verify that the following are present:

a. `com.liferay.registry.api.jar`

b. portal-kernel.jar
c. portlet.jar
d. (database of your choice jar) (e.g. MySQL, SQL Server...)

2. Inside the osgi folder, verify the following folders are present:

    a. configs
    b. core
    c. marketplace
    d. target-platform
    e. test

Running Liferay DXP on JBoss EAP 6.4 in Standalone Mode vs. Domain Mode

JBoss EAP 6.4 can be launched in either *standalone* mode or *domain* mode. Domain mode allows multiple application server instances to be managed from a single control point. A collection of such application servers is known as a *domain*. For more information on standalone mode vs. domain mode, please refer to the section on this topic in the JBoss EAP 6.4 Administration and Configuration Guide. Liferay DXP fully supports JBoss EAP 6.4 when it runs in standalone mode but not when it runs in domain mode.

You can run Liferay DXP on JBoss EAP 6.4 in domain mode, but this method is not fully supported. In particular, Liferay DXP's hot-deploy does not work, since JBoss EAP 6.4 cannot deploy non-exploded .war files in domain mode. Instead, .war files are in the domain/data/content directory. Deployments are only possible using the command line interface. This prevents many Liferay DXP plugins from working as intended. For example, JSP hooks don't work on JBoss EAP 6.4 running in domain mode, since Liferay DXP's JSP override mechanism relies on the application server reloading customized JSP files from the exploded plugin .war file location. Other plugins, such as service or action hooks, should still work properly since they don't require JBoss to access anything (such as JSP files) from an exploded .war file on the file system.

Note: This does not prevent Liferay DXP from running in a clustered environment on multiple JBoss servers. You can set up a cluster of Liferay DXP instances running on JBoss EAP 6.4 servers running in standalone mode. Please refer to the chapter of this guide on Configuring Liferay DXP for High Availability for information on setting up a Liferay DXP cluster.

Configuring JBoss

Now you'll make some adjustments in your configuration to support using Liferay DXP.

You can specify the JBoss server instance's configuration in the XML file $JBOSS_HOME/standalone/configuration/standalone
You must also make some modifications to your configuration and startup scripts found in the $JBOSS_HOME/bin/ folder. Lastly, you'll need to make some modifications in your $JBOSS_HOME/modules/. You'll begin with making changes to standalone.xml.

Make the following modifications to standalone.xml:

1. In the <jsp-configuration> tag, set the Java VM compatibility for Liferay source and class files. They are compatible with Java 8 by default.

```
<jsp-configuration development="true" source-vm="1.8" target-vm="1.8" />
```

2. Locate the closing </extensions> tag. Directly beneath that tag, insert the following system properties:

```
<system-properties>
    <property name="org.apache.catalina.connector.URI_ENCODING" value="UTF-8" />
    <property name="org.apache.catalina.connector.USE_BODY_ENCODING_FOR_QUERY_STRING" value="true" />
</system-properties>
```

3. Add a timeout for the deployment scanner by setting `deployment-timeout="360"` as seen in the excerpt below.

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
    <deployment-scanner deployment-timeout="360" path="deployments" relative-to="jboss.server.base.dir" scan-interval="5000"/>
</subsystem>
```

4. Add the following JAAS security domain to the security subsystem `<security-domains>` defined in element `<subsystem xmlns="urn:jboss:domain:security:1.2">`.

```
<security-domain name="PortalRealm">
    <authentication>
        <login-module code="com.liferay.portal.kernel.security.jaas.PortalLoginModule" flag="required" />
    </authentication>
</security-domain>
```

5. Disable the default JBoss Welcome page by setting the `enable-welcome-root` attribute to `false`, as seen in the snippet below.

```
<subsystem xmlns="urn:jboss:domain:web:2.2" default-virtual-server="default-host" native="false">
    <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
    <virtual-server name="default-host" enable-welcome-root="false">
    ...
```

6. In the same `<subsystem ... />` element that was outlined in the previous step, add the following snippet above the `<connector ... />` element:

```
<configuration>
    <jsp-configuration source-vm="1.8" target-vm="1.8" development="true" />
</configuration>
```

Checkpoint:

1. The `standalone.xml` has been modified with the following values:
    a. `deployment-timeout="360"` has been set. The value is in seconds.
    b. The JAAS security domain has been added.
    c. The property `enable-welcome-root="false"` has been set.
    d. The property `jsp-configuration source-vm="1.8" target-vm="1.8" development="true"` has been set in the `<configuration>` element. This is required because Liferay DXP runs on Java JDK 1.8 or else there will be a JasperException and the instance will fail to start.

Now it's time for some changes to your configuration and startup scripts.

Make the following modifications to your standalone domain's configuration script file `standalone.conf` (`standalone.conf.bat` on Windows) found in your `$JBOSS_HOME/bin/` folder.

These modifications change the following options: - Set the file encoding - Set the user time-zone - Set the preferred protocol stack - Increase the default amount of memory available.

Make the following edits as applicable to your operating system:

On Windows, comment out the initial `JAVA_OPTS` assignment as demonstrated in the following line:

```
rem set "JAVA_OPTS=-Xms1G -Xmx1G -XX:MaxPermSize=256M"
```

Then add the following `JAVA_OPTS` assignment one line above the `:JAVA_OPTS_SET` line found at end of the file:

```
set "JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8 -Djava.net.preferIPv4Stack=true -Dsecmgr -Djava.security.policy=$JBOSS_HOME/bin/server.policy -
Djboss.home.dir=$JBOSS_HOME -Duser.timezone=GMT -Xmx1024m -XX:MaxMetaspaceSize=384m"
```

On Unix, merge the following values into your settings for `JAVA_OPTS`, replacing any matching attributes with the ones found in the assignment below:

```
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF-8 -Djava.net.preferIPv4Stack=true -Dsecmgr -Djava.security.policy=$JBOSS_HOME/bin/server.policy -
Djboss.home.dir=$JBOSS_HOME -Duser.timezone=GMT -Xmx1024m -XX:MaxMetaspaceSize=384m"
```

Important: For Liferay DXP to work properly, the application server JVM must use the `GMT` time zone and `UTF-8` file encoding.

Make sure you replace the `$JBOSS_HOME` references with the appropriate directory. You'll notice some Java security options. You'll finish configuring the Java security options in the *Security Configuration* section.

Note: If you plan on using the IBM JDK with your JBoss server, you'll need to complete some additional steps. First, navigate to the `$JBOSS_HOME/modules/com/liferay/portal/main/module.xml` file and insert the following dependency within the `<dependencies>` element:

```
<module name="ibm.jdk" />
```

Then navigate to the `$JBOSS_HOME/modules/system/layers/base/sun/jdk/main/module.xml` file and insert the following path names inside the `<paths>...</paths>` element:

```
<path name="com/sun/crypto" />
<path name="com/sun/crypto/provider" />
<path name="com/sun/image/codec/jpeg" />
<path name="com/sun/org/apache/xml/internal/resolver" />
<path name="com/sun/org/apache/xml/internal/resolver/tools" />
```

The added paths resolve issues with portal deployment exceptions and image uploading problems on a Liferay DXP instance running on JBoss EAP 6.4.

Checkpoint:

1. The `standalone.conf.bat` file has been updated with the following changes:
    a. UTF-8 file encoding
    b. The user time-zone
    c. The preferred protocol stack
    d. Increased the default amount of memory available.
2. If using the IBM JDK with the JBoss server, `<module name="ibm.jdk" />` has been added to the `$JBOSS_HOME/modules/com/liferay/portal/main/module.xml` file.
3. The additional IBM JDK paths have been set in the `$JBOSS_HOME/modules/system/layers/base/sun/jdk/main/module.xml` file.

The prescribed script modifications are now complete for your Liferay DXP installation on JBoss. Next you'll configure mail and the database.

Database Configuration

If you want JBoss to manage your data source, follow the instructions in this section. If you want to use the built-in Liferay DXP data source, you can skip this section.

Modify `standalone.xml` and add your data source and driver in the `<datasources>` element of your data sources subsystem.

1. First, add your data source inside the `<datasources>` element.

```
 <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" jta="true" use-java-context="true" use-
ccm="true">
     <connection-url>jdbc:mysql://localhost/lportal</connection-url>
     <driver>mysql</driver>
     <security>
         <user-name>root</user-name>
         <password>root</password>
     </security>
 </datasource>
```

   Be sure to replace the database name (i.e. `lportal`), user name, and password with the appropriate values.

```
**Note:** If you'd like to change your datasource `jndi-name` to something
different, you'll need to also edit the `datasource` element in the
`<default-bindings>` tag.
```

---

2. Add your driver to the `<drivers>` element also found within the `<datasources>` element.

```
 <drivers>
     <driver name="mysql" module="com.liferay.portal"/>
 </drivers>
```

   Your final data sources subsystem should look like this:

```
<subsystem xmlns="urn:jboss:domain:datasources:1.2">
    <datasources>
        <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" jta="true" use-java-
context="true" use-ccm="true">
            <connection-url>jdbc:mysql://localhost/lportal</connection-url>
            <driver>mysql</driver>
            <security>
                <user-name>root</user-name>
                <password>root</password>
            </security>
        </datasource>
        <drivers>
            <driver name="mysql" module="com.liferay.portal"/>
        </drivers>
    </datasources>
</subsystem>
```

   Now that you've configured your data source, the mail session is next.

Mail Configuration

If you want JBoss to manage your mail session, use the following instructions. If you want to use the built-in Liferay DXP mail session, you can skip this section.

Specify your mail subsystem in `standalone.xml` as in the following example:

```
<subsystem xmlns="urn:jboss:domain:mail:1.2">
    <mail-session jndi-name="java:jboss/mail/MailSession" >
        <smtp-server ssl="true" outbound-socket-binding-ref="mail-smtp">
            <login username="USERNAME" password="PASSWORD"/>
        </smtp-server>
   </mail-session>
</subsystem>
...
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.binding.port-offset:0}">
...
<outbound-socket-binding name="mail-smtp">
        <remote-destination host="smtp.gmail.com" port="465"/>
    </outbound-socket-binding>
</socket-binding-group>
```

You've got mail! Next, you'll make sure Liferay DXP can connect using your new mail session and database.

Configuring data sources and mail sessions

Now that your data source and mail session are set up, you need to ensure 7.0 can access them.

1. First, navigate to the Liferay Home folder, which is one folder above JBoss's install location (i.e. `$JBOSS_HOME/..`).

2. If you're using *JBoss* to manage your data source, add the following configuration to your `portal-ext.properties` file in your *Liferay Home* to refer to your data source:

   ```
   jdbc.default.jndi.name=java:jboss/datasources/ExampleDS
   ```

   If you're using *7.0* to manage your data source, follow the instructions for using the setup wizard.

3. If you're using *7.0* to manage your mail session, this configuration is done in 7.0. That is, after starting your portal as described in the *Deploy Liferay DXP* section, go to *Control Panel* → *Server Administration* → *Mail* and enter the settings for your mail session.

   If you're using JBoss to manage your mail session, add the following configuration to your `portal-ext.properties` file to reference that mail session:

   ```
   mail.session.jndi.name=java:jboss/mail/MailSession
   ```

Before you deploy 7.0 on your JBoss app server, you should enable and configure Java security so you can use Liferay DXP's plugin security manager with your downloaded Liferay DXP applications.

Security Configuration

When you're ready to begin using other people's apps from Marketplace, you'll want to protect your Liferay DXP instance and your JBoss server from security threats. To do so, you can enable Java Security on your JBoss server and specify a security policy to grant your Liferay DXP instance access to your server.

Remember, you set the `-Dsecmgr` and `-Djava.security.policy` Java options in the `standalone.conf.bat` file earlier in the *Configuring JBoss* section. The `-Dsecmgr` Java option enables security on JBoss. Likewise, the `-Djava.security.policy` Java option lists the permissions for your server's Java security policy. If you have not set these options, you'll need to do so before using Java security.

This configuration opens up all permissions. You can tune the permissions in your policy later. Create the `$JBOSS_HOME/bin/server.policy` file and add the following contents:

```
grant {
    permission java.security.AllPermission;
};
```

For extensive information on Java SE Security Architecture, see its specification documents at http://docs.oracle.com/javase/7/docs/technotes/guides/security/spec/security-spec.doc.html. Also, see the Plugin Security and PACL tutorial to learn how to configure Liferay DXP plugin access to resources.

Deploy Liferay DXP

1. If the folder `$JBOSS_HOME/standalone/deployments/ROOT.war` already exists in your JBoss installation, delete all of its subfolders and files. Otherwise, create a new folder `$JBOSS_HOME/standalone/deployments/ROOT.war`.
2. Unzip the Liferay DXP `.war` file into the `ROOT.war` folder.
3. To trigger deployment of `ROOT.war`, create an empty file named `ROOT.war.dodeploy` in your `$JBOSS_HOME/standalone/deployments/` folder. On startup, JBoss detects the presence of this file and deploys it as a web application.
4. Start the JBoss application server by navigating to `$JBOSS_HOME/bin` and running `standalone.bat` or `standalone.sh`.

The JBoss application server starts and deploys Liferay DXP.

Warning: The JBoss application server system property `jboss.as.management.blocking.timeout` specifies an application container stability timeout (the default is `300` seconds). If container stability times out during startup, all applications are undeployed and the container shuts down. The error message looks like this:

```
12:21:13,956 ERROR [org.jboss.as.controller.management-operation] (Controller Boot Thread) JBAS013412: Timeout after [300] seconds waiting for service cont
```

JBoss CLI, Java options, and `.conf` files let you modify the timeout (e.g., increase the timeout to give the container more time to stabilize). Here's a `900` second timeout set in a `.conf` file property.

```
...
</extensions>
<system-properties>
      <property name="jboss.as.management.blocking.timeout" value="900"/>
</system-properties>
<management>
...
```

JBoss documentation has more details.

You're now an expert when it comes to deploying Liferay DXP on JBoss!

36.7 Installing Liferay DXP on tc Server

Liferay DXP is supported on tc Server. Please see the Compatibility Matrix for the supported version. Before proceeding, you must download these *Additional Files*:

- Liferay DXP WAR file
- Dependencies ZIP file
- OSGi Dependencies ZIP file

Once you have those pieces of the puzzle, you just need to assemble them.
Installing Liferay DXP manually requires these basic steps:

- Installing Liferay DXP dependencies to your application server
- Configuring your application server for Liferay DXP
- Installing Liferay DXP by providing the WAR file to your application server and the OSGi folder for Liferay DXP

Note: You'll see the term *Liferay Home* used in this installation guide. *Liferay Home* refers to the folder containing your tc Server instance and some Liferay DXP-specific folders:. `data`, `deploy`, `licenses`, and `osgi` folders.

Installing Liferay DXP Dependencies

Liferay DXP depends on some additional JARs that aren't included with tc Server by default. There are even more JARs that you'd find in a Liferay DXP bundle that are not required but can be useful. If you don't have a Liferay DXP bundle, you can download the required JARs from third parties, as described below.

Note: Many required and useful JARs are pre-installed when you build Liferay DXP from the source code or download a Liferay DXP bundle. If you want to acquire all of the JARs that ship with a Liferay DXP bundle quickly, using one of these sources might save you time.

Here are the JARs included in the dependencies zip file:

- `com.liferay.registry.api-1.0.4.jar`
- `hsql.jar`
- `portal-kernel.jar`
- `portlet.jar`

One JAR you definitely need that is not included in the dependencies zip is your database driver. Database drivers for *MySQL* and *PostgreSQL* can be found in a Liferay DXP bundle or in the Liferay DXP source code.

There are several other dependency JARs that aren't included in the dependencies zip. If you don't have them already on hand or have access to a Liferay DXP bundle, you'll have to download them yourself.

- `jta.jar`: Support for Java transactions. You can get this `.jar`, which manages transactions, from http://www.oracle.com/technetwork/java/javaee/jta/index.html

- `mail.jar`: Support for the Java Mail API. You can get this `.jar` from http://www.oracle.com/technetwork/java/index-138643.html
- `persistence.jar`: Support for the Java Persistence API. You can get this `.jar` from http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html
- `activation.jar`: This is an implementation of the Java Activation Framework. You can get this `.jar` from http://www.oracle.com/technetwork/java/jaf11-139815.html
- `ccpp.jar`: Enables Composite Capability/Preference Profiles. You can get this `.jar` from http://mvnrepository.com/artifact/javax.ccpp/ccpp/1.0
- `jms.jar`: The Java Messaging Service. You can get this `.jar` from http://www.oracle.com/technetwork/java/docs-136352.html
- `jutf7.jar`: Provides UTF-7 and Modified UTF-7 charsets for Java. You can get this `.jar` from http://sourceforge.net/projects/jutf7/
- `junit.jar`: Lets you run unit tests. You can get this `.jar` from http://sourceforge.net/projects/junit/

Place these JARs in *Liferay Home*'s `lib` folder (not tc Server's) (see more steps below).

Liferay DXP includes an OSGi runtime. Extract the OSGi ZIP file that you downloaded and copy the `osgi` folder to your Liferay Home folder. The `osgi` folder contains many required JAR files and a few configuration files.

Configuring tc Server

There are a few configuration edits to make so Liferay DXP runs well on tc Server. All of these configuration changes should be made in your tc Server runtime instance.

1. Download and unzip tc Server available here. This will be referred to as `[TCSERVER_INSTANCE_HOME]`.
2. Create a folder called `servers` inside `[TCSERVER_INSTANCE_HOME]`. (e.g. `/opt/pivotal-tc-server-standard-3.1.8.release/servers`). This folder becomes *Liferay Home* (see note above) and you should not confuse the two.
3. Next, create an instance called *dxp-server* where Liferay DXP will be deployed. Run the command:

```
tcruntime-instance.bat|sh create -i servers dxp-server
```

4. Copy the dependencies `jars` inside the `lib` folder inside `[TCSERVER_INSTANCE_HOME]/servers/dxp-server`.

Checkpoint:

1. A new folder called `servers` has been created.
2. A new folder called `dxp-server` has been created inside the `servers` folder. The following folders have been created inside the `dxp-server` folder:

- `bin`
- `conf`

- lib
- logs
- temp
- webapps
- work

Liferay DXP dependencies have been placed inside the [TCSERVER_INSTANCE_HOME]/servers/dxp-server/lib folder:

- com.liferay.registry.api-1.0.4.jar
- hsql.jar
- portal-kernel.jar
- portlet.jar
- jta.jar
- junit.jar
- jutf7.jar
- jms.jar
- mail.jar
- persistence.jar
- activation.jar
- ccp.jar
- a database jar for other than HSQL (e.g. mariadb, mysql, db2)

There are a few more configuration changes for Liferay DXP to run well on tc Server 3. All of these configuration changes should be made in the tc Server runtime instance.

1. Navigate to the [TCSERVER_INSTANCE_HOME]/servers/dxp-server/bin folder. In setenv.sh replace this line

   ```
   JVM_OPTS="-Xmx512M -Xss256K"
   ```

   with this one

   ```
   JVM_OPTS="-Dfile.encoding=UTF-8 -Duser.timezone=GMT -Xmx1024M -Xss512K -XX:MaxMetaspaceSize=512m"
   ```

   In setenv.bat replace

   ```
   set JVM_OPTS=-Xmx512M -Xss256K
   ```

   with

   ```
   set JVM_OPTS=-Dfile.encoding=UTF-8 -Duser.timezone=GMT -Xmx1024M -Xss512K -XX:MaxMetaspaceSize=512m
   ```

```
**Important:** For Liferay DXP to work properly, the application server JVM must
use the `GMT` time zone and `UTF-8` file encoding.
```

---

2. Next, you should make sure that UTF-8 URI encoding is used consistently. Open `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/conf/server.xml` and make sure the `Connector` tag includes setting the `URIEncoding` to `UTF-8`.

```
<Connector acceptCount="100"
           connectionTimeout="20000"
           executor="tomcatThreadPool"
           maxKeepAliveRequests="15"
           port="${bio.http.port}"
           protocol="org.apache.coyote.http11.Http11Protocol"
           redirectPort="${bio.https.port}"
           URIEncoding="UTF-8" />
```

3. If you're installing Liferay DXP and tc Server on Windows, open `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/conf/wrapper.conf` and replace

```
wrapper.java.additional.8=-Xmx512M
```

with

```
wrapper.java.additional.8="-Xmx1024M"
```

and

```
wrapper.java.additional.9=-Xss256K
```

with

```
wrapper.java.additional.9="-Xss512K"
wrapper.java.additional.10="-XX:MaxMetaspaceSize=256M"
wrapper.java.additional.11="-Dfile.encoding=UTF-8"
```

```
**Important:** For Liferay DXP to work properly, the application server JVM
must use the `GMT` time zone and `UTF-8` file encoding. If your Java wrapper
doesn't already specify the `GMT` time zone, add an entry for it:

    wrapper.java.additional.12=-Duser.timezone=GMT
```

---

4. Last, open `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/conf/web.xml` and add the following after `<load-on-startup>3</load-on-startup>`

```
<init-param>
    <param-name>compilerSourceVM</param-name>
    <param-value>1.8</param-value>
</init-param>
<init-param>
    <param-name>compilerTargetVM</param-name>
    <param-value>1.8</param-value>
</init-param>
```

Database Configuration

The easiest way to handle your database configuration is to let Liferay DXP manage your data source. If you want to use Liferay DXP's built-in data source, you can skip this section.

If you want tc Server to manage your data source, use this procedure:

1. Make sure your database server is installed and working. If it's installed on a different machine, make sure it's accessible from your Liferay DXP machine.

2. Add your data source as a resource in the context of your web application specified in `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/conf/Catalina/localhost/ROOT.xml` (create this file if you don't have it already):

```
<Resource
    name="jdbc/LiferayPool"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost/lportal?useUnicode=true&amp;characterEncoding=UTF-8"
    username="root"
    password="root"
    maxActive="100"
    maxIdle="30"
    maxWait="10000"
/>
```

Note that the above resource definition assumes your database name is *lportal*, that you're using MySQL, and that your MySQL *username* and *password* are both *root*. You'll have to update these values with your own database name and credentials.

Your data source is now configured. Next set up the mail session.

Mail Configuration

As with database configuration, the easiest way to handle mail configuration is to let Liferay DXP handle your mail session. If you want to use Liferay DXP's built-in mail session, skip this section and use Liferay DXP's Control Panel to configure a mail server after Liferay DXP has been installed and started.

To create a mail session bound to `mail/MailSession`, edit `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/conf/Catalina/localhost/ROOT.xml` and configure your mail session in a `Resource` tag. Make sure to replace the example mail session values with your own.

```
<Resource
    name="mail/MailSession"
    auth="Container"
    type="javax.mail.Session"
    mail.pop3.host="pop.gmail.com"
    mail.pop3.port="110"
    mail.smtp.host="smtp.gmail.com"
    mail.smtp.port="465"
    mail.smtp.user="user"
    mail.smtp.password="password"
    mail.smtp.auth="true"
    mail.smtp.starttls.enable="true"
    mail.smtp.socketFactory.class="javax.net.ssl.SSLSocketFactory"
    mail.imap.host="imap.gmail.com"
    mail.imap.port="993"
    mail.transport.protocol="smtp"
    mail.store.protocol="imap"
/>
```

Your mail session is configured. Next, you'll make sure Liferay DXP can access your mail session and database.

Configuring a Database and Mail Session

In this section you'll specify appropriate properties for Liferay DXP to use in connecting to your database and mail session.

1. If you are using tc Server to manage your data source, add the following line to the `portal-ext.properties` file in your *Liferay Home*.

   ```
   jdbc.default.jndi.name=jdbc/LiferayPool
   ```

2. If you are using tc Server to manage your mail session, add the following to `portal-ext.properties`:

   ```
   mail.session.jndi.name=mail/MailSession
   ```

Before you deploy Liferay DXP, you should configure a Portal Access Control List (PACL).

Enabling PACL

To enable PACL, you need to enable Tomcat's security manager. Make sure `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/conf/catalina.policy` specifies the permissions

```
grant {
    permission java.security.AllPermission;
};
```

Open `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/bin/setenv.sh` if on Linux, Unix, or Mac OS, or `setenv.bat` if on Windows. Enable the security manager by inserting the following code into the `CATALINA_OPTS` variable (inside quotation marks):

```
-Djava.security.manager -Djava.security.policy==$CATALINA_BASE/conf/catalina.policy
```

Now you have PACL enabled and configured for your portal.

Deploying Liferay DXP

Now you're ready to deploy Liferay DXP using your Liferay DXP WAR file.

1. If you are manually installing Liferay DXP on a clean tc Server instance, delete the contents of the `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/webapps/ROOT` directory. This removes the default Tomcat home page.

2. Extract the Liferay DXP `.war` file to `[TCSERVER_INSTANCE_HOME]/servers/dxp-server/webapps/ROOT`.

   Now it's time to launch Liferay DXP!

3. Start tc Server by navigating to the runtime instance's `bin` folder and executing `./tcruntime-ctl.sh run` or `/tcruntime-ctl.sh start`. If you're on Windows you can use the `.bat` version of the startup script only if you installed the tc Server instance as a Windows service. Alternatively, you can use `./catalina.sh run` or `catalina.bat run`.

After you set up your account you can start using Liferay DXP.

36.8 Installing Liferay DXP on WebLogic 12c R2

Although it's possible to install Liferay DXP in a WebLogic Admin Server, this isn't recommended. It's best practice to install web apps, including Liferay DXP, in a WebLogic Managed server. By deploying to a Managed Server, you'll be able to start/shutdown Liferay DXP more quickly, and you'll more easily be able to extend Liferay DXP into a cluster configuration. This article therefore focuses on installing Liferay DXP in a Managed Server.

Before getting started, you should take care of a few things. First, it's assumed that your Admin and Managed Servers already exist. See WebLogic's documentation for instructions on setting up and configuring Admin and Managed Servers.

You should also read the following articles to familiarize yourself with Liferay DXP's general installation steps:

- Preparing for install
- Installing Liferay DXP
- Installing Liferay DXP Manually

Before proceeding, you must download these *Additional Files*:

- Liferay DXP WAR file
- Dependencies ZIP file
- OSGi Dependencies ZIP file

Without any further ado, get ready to install Liferay DXP in WebLogic!

Configuring WebLogic's Node Manager

WebLogic's Node Manager starts and stops managed servers.

If you're running WebLogic on a UNIX system other than Solaris or Linux, use the Java Node Manager, instead of the native version of the Node Manager, by configuring these Node Manager properties in the `domains/your_domain_name/nodemanager/nodemanager.properties` file:

```
NativeVersionEnabled=false

StartScriptEnabled=true
```

Note: By default, SSL is used with Node Manager. If you want to disable SSL during development, for example, set `SecureListener=false` in your `nodemanager.properties` file.

See Oracle's Configuring Java Node Manager documentation for details.

Configuring WebLogic

Configure the JVM and other options in a `setUserOverridesLate` WebLogic startup script and in your Managed Server UI.

Create a `setUserOverridesLate.sh` script in `[Your Domain]/bin` and add the following settings.

```
export DERBY_FLAG="false"
export JAVA_OPTIONS="${JAVA_OPTIONS} -Dfile.encoding=UTF-8 -Duser.timezone=GMT -da:org.apache.lucene... -da:org.aspectj..."
export JAVA_PROPERTIES="-Dfile.encoding=UTF-8 ${JAVA_PROPERTIES} ${CLUSTER_PROPERTIES}"
export MW_HOME="[place your WebLogic Server folder path here]"
export USER_MEM_ARGS="-Xmx1024m -XX:MetaspaceSize=512m"
```

Important: For Liferay DXP to work properly, the application server JVM must use the `GMT` time zone and `UTF-8` file encoding.

The `DERBY_FLAG` setting disables the Derby server built in to WebLogic, as DXP does not require this server.

`JAVA_OPTIONS` sets DXP's UTF-8 requirement, Lucene usage, and Aspect Oriented Programming via AspectJ.

`JAVA_PROPERTIES` also sets DXP's UTF-8 requirement.

Set `MW_HOME` to the folder containing the WebLogic server on the machine. For example,

```
export MW_HOME="/Users/ray/Oracle/wls12210"
```

You must also ensure that the Node Manager sets Liferay DXP's memory requirements when starting the Managed Server. In the Admin Server's console UI, navigate to the Managed Server you want to deploy Liferay DXP to and select the *Server Start* tab. Enter the following into the *Arguments* field:

```
-Xmx2048m -XX:MaxMetaspaceSize=512m
```

Click *Save* when you're finished.

Next, you'll set some Liferay DXP-specific properties for your Liferay DXP installation.

Setting Liferay DXP Properties

Before installing Liferay DXP, you must set the *Liferay Home* folder's location via the `liferay.home` property in a `portal-ext.properties` file. You can also use this file to override other Liferay DXP properties that you may need.

First, decide which directory you want to serve as Liferay Home. In WebLogic, your domain's folder is generally Liferay Home, but you can choose any folder on your machine. Then create your `portal-ext.properties` file and add the `liferay.home` property:

```
liferay.home=/full/path/to/your/liferay/home/folder
```

Remember to change this file path to the location on your machine that you want to serve as Liferay Home.

Now that you've created your `portal-ext.properties` file, you must put it inside the Liferay DXP WAR file. Expand the Liferay DXP WAR file and place `portal-ext.properties` in the `WEB-INF/classes` folder. Later, you can deploy the expanded archive to WebLogic. Alternatively, you can re-WAR the expanded archive for later deployment. In either case, Liferay DXP reads your property settings once it starts up.

If you need to make any changes to `portal-ext.properties` after Liferay DXP deploys, you can find it in your domain's `autodeploy/ROOT/WEB-INF/classes` folder. Note that the `autodeploy/ROOT` folder contains the Liferay DXP deployment.

Next, you'll install Liferay DXP's dependencies.

Installing Liferay DXP Dependencies

You must now install Liferay DXP's dependencies. Recall that earlier you downloaded two ZIP files containing these dependencies. Install their contents now:

1. `liferay-dxp-digital-enterprise-dependencies-[version].zip`: Unzip this file and place its contents in your WebLogic domain's `lib` folder.

2. `liferay-dxp-digital-enterprise-osgi-[version].zip`: Unzip this file and place its contents in the `Liferay_Home/osgi` folder (create this folder if it doesn't exist).

If you don't want to use Liferay DXP's built-in Hypersonic database, you must also add your database's driver JAR file to your domain's `lib` folder. Note that although Hypersonic is fine for testing purposes, you **should not** use it for production Liferay DXP instances.

Next, you'll configure your database.

Database Configuration

Use the following procedure if you want WebLogic to manage your database for Liferay DXP. You can skip this section if you want to use Liferay DXP's built-in Hypersonic database.

1. Log in to your AdminServer console.
2. In the *Domain Structure* tree, find your domain and navigate to *Services* → *JDBC* → *Data Sources*.
3. To create a new data source, click *New*. Fill in the *Name* field with `Liferay Data Source` and the *JNDI Name* field with `jdbc/LiferayPool`. Select *MySQL* as the database type, and *MySQL's Driver (Type 4) Versions:using com.mysql.jdbc.Driver* as the database driver. Click *Next* to continue.
4. Accept the default settings on this page and click *Next* to move on.
5. Fill in the database information for your MySQL database.
6. Add the text `?useUnicode=true&characterEncoding=UTF-8&\useFastDateParsing=false` to the URL line and test the connection. If it works, click *Next*.
7. Select the target for the data source and click *Finish*.
8. You must now tell Liferay DXP about the JDBC data source. Create a `portal-ext.propreties` file in your Liferay Home directory, and add the line `jdbc.default.jndi.name=jdbc/LiferayPool`.

Alternatively, you can make the above configuration strictly via properties in the `portal-ext.properties` file. To do so, place the following properties and values in the file. Be sure to change the `your*` values with the values appropriate for your database's configuration:

```
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.url=jdbc:mysql://your.db.ip.address/yourdbname?useUnicode?useUnicode=true&characterEncoding=UTF-8&useFastDateParsing=false
jdbc.default.username=yourdbuser
jdbc.default.password=yourdbpassword
```

Next, you'll configure your mail session.

Mail Configuration

If you want WebLogic to manage your mail session, use the following procedure. If you want to use Liferay's built-in mail session (recommended), you can skip this section.

1. Start WebLogic and log in to your Admin Server's console.
2. Select *Services* → *Mail Sessions* from the *Domain Structure* box on the left hand side of your Admin Server's console UI.

3. Click *New* to begin creating a new mail session.

4. Name the session *LiferayMail* and give it the JNDI name `mail/MailSession`. Then fill out the *Session Username*, *Session Password*, *Confirm Session Password*, and *JavaMail Properties* fields as necessary for your mail server. See the WebLogic documentation for more information on these fields. Click *Next* when you're done.

5. Choose the Managed Server that you'll install Liferay DXP on, and click *Finish*. Then shut down your Managed and Admin Servers.

6. With your Managed and Admin servers shutdown, add the following property to your `portal-ext.properties` file:

   ```
   mail.session.jndi.name=mail/MailSession
   ```

   Liferay DXP references your WebLogic mail session via this property setting. If you've already deployed Liferay DXP, you can find your `portal-ext.properties` file in your domain's `autodeploy/ROOT/WEB-INF/classes` folder.

Your changes will take effect upon restarting your Managed and Admin servers.
Next, you'll configure the security settings in your WebLogic server to work with Liferay DXP.

Security Configuration

When you're ready to start using Liferay Marketplace apps, you'll want to protect your Liferay DXP instance and your WebLogic server from security threats. To do so, you must enable Java Security on your WebLogic server and specify a security policy to grant Liferay DXP access to your server.

First, you'll grant Liferay DXP access to your server. This configuration opens all permissions–you can fine-tune your policy's permissions later. If it doesn't already exist, create a policy file named `weblogic.policy` in your `$WL_HOME/server/lib` folder. Replace its contents with the following:

```
grant {
    permission java.security.AllPermission;
};
```

To enable security on your WebLogic server and direct the server to your policy file, open the `setDomainEnv.[cmd|sh]` file in your domain's `bin`folder. Then set the `-Djava.security.manager` Java option and set the property `-Djava.security.policy` to your `weblogic.policy` file. You can specify both settings on the same line like this:

```
-Djava.security.manager -Djava.security.policy==$WL_HOME/server/lib/weblogic.policy
```

The double equals sign tells the app server to use only this policy file. Any other security policies are ignored.

For extensive information on Java SE Security Architecture, see its specification docs at http://docs.oracle.com/javase/7/docs/technotes/guides/security/spec/security-spec.doc.html. Also, see the section Understanding Plugin Security Management in the Developer's Guide to learn how to configure Liferay DXP plugin access to resources.

Deploying Liferay DXP

As mentioned earlier, although you can deploy Liferay DXP to a WebLogic Admin Server, you should instead deploy it to a WebLogic Managed Server. Dedicating the Admin Server to managing other servers that run your apps is a best practice.

Follow these steps to deploy Liferay DXP to a Managed Server:

1. Make sure the Managed Server you want to deploy Liferay DXP to is shut down.
2. In your Admin Server's console UI, select *Deployments* from the *Domain Structure* box on the left hand side. Then click *Install* to start a new deployment.
3. Select the Liferay DXP WAR file or its expanded contents on your file system. Alternatively, you can upload the WAR file by clicking the *Upload your file(s)* link. Click *Next*.
4. Select *Install this deployment as an application* and click *Next*.
5. Select the Managed Server you want to deploy Liferay DXP to, and click *Next*.
6. If the default name is appropriate for your installation, keep it. Otherwise, give it a name of your choosing and click *Next*.
7. Click *Finish*. After the deployment finishes, click *Save* if you want to save the configuration.
8. Start the Managed Server you deployed Liferay DXP to. Liferay DXP precompiles all the JSPs, and then launches.

Nice work! Now you're running Liferay DXP on WebLogic.

36.9 Installing Liferay DXP on WebSphere 8.5.5

IBM ® WebSphere ® is a trademark of International Business Machines Corporation, registered in many jurisdictions worldwide.

Tip: Throughout this installation and configuration process, WebSphere prompts you to click *Save* to apply changes to the Master Configuration. Do so intermittently to save your changes.

For Liferay DXP to work correctly, WebSphere 8.5.5 Fix Pack 11 (or later) must be installed. You can find more information about this fix pack here.

Please also note that Liferay DXP doesn't support the WebSphere Application Liberty Profile.

You should also read the following articles to familiarize yourself with Liferay DXP's general installation steps:

- Preparing for install
- Installing Liferay DXP
- Installing Liferay DXP Manually

Note that the *Liferay Home* folder is important to the operation of Liferay DXP. In Liferay Home, Liferay DXP creates certain files and folders that it needs to run. On WebSphere, Liferay Home is typically `[Install Location]/WebSphere/AppServer/profiles/your-profile/liferay`.

Before proceeding, you must download these *Additional Files*:

- Liferay DXP WAR file
- Dependencies ZIP file
- OSGi Dependencies ZIP file

Without any further ado, get ready to install Liferay DXP in WebSphere!

Preparing WebSphere for Liferay DXP

When the application server binaries have been installed, start the *Profile Management Tool* to create a profile appropriate for Liferay DXP.

1. Click *Create...*, choose *Application Server*, and then click *Next*.

2. Click the *Advanced* profile creation option and then click *Next*. Why Advanced? You can specify your own values for settings such as the location of the profile and names of the profile, node and host. You can assign your own ports. You can optionally choose whether to deploy the administrative console and sample application and also add web-server definitions if you wish. Web server definitions are used with IBM HTTP Server. See the WebSphere documentation for more information about these options.

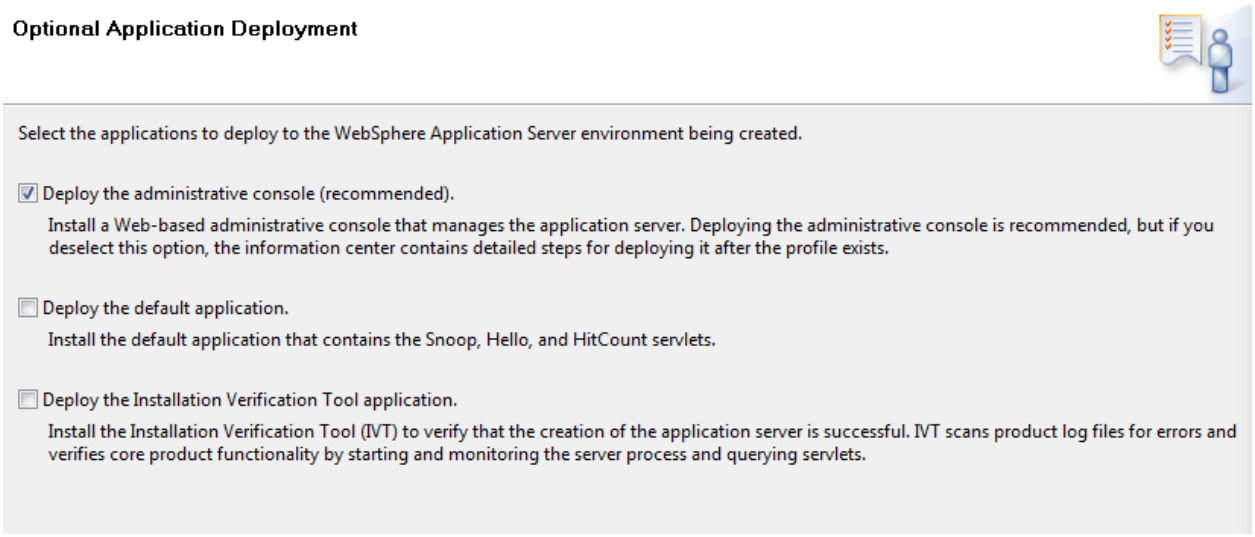


Figure 36.2: Choose the Advanced profile option to specify your own settings.

3. Check the box *Deploy the administrative console*. This gives you a web-based UI for working with your application server. Skip the default applications. You'd only install these on a development machine. Click *Next*.

4. Set the profile name and location. Ensure you specify a performance tuning setting other than *Development*, since you're installing a production server. See the WebSphere documentation for more information about performance tuning settings. Click *Next*.

5. Choose node, server, and host names for your server. These will be specific to your environment. Click *Next*.

6. Administrative security in WebSphere is a way to restrict who has access to the administrative tools. You may want to have it enabled in your environment so that a user name and password are required to administer the WebSphere server. See WebSphere's documentation for more information. Click *Next*.

7. Each profile needs a security certificate, which comes next in the wizard. If you don't have certificates already, choose the option to generate a personal certificate and a signing certificate and click *Next*.

8. Once the certificates are generated, set a password for your keystore. Click *Next*.

9. Now you can customize the ports this server profile uses. Be sure to choose ports that are open on your machine. When choosing ports, the wizard detects existing WebSphere installations and if it finds activity, it increments ports by one.

10. Choose whether you want this profile started when the machine starts. Click *Next*.

11. WebSphere ships with IBM HTTP Server, which is a re-branded version of Apache. Choose whether you want a web server definition, so that this JVM receives requests forwarded from the HTTP server. See WebSphere's documentation for details on this. When finished, click *Next*.

12. The wizard then shows you a summary of what you selected, enabling you to keep your choices or go back and change something. When you're satisfied, click *Next*.

WebSphere then creates your profile and finishes with a message telling you the profile was created successfully. Awesome! Your profile is complete. Now there are a few things you need to configure in your application server.

Configuring the WebSphere Application Server

In this version of WebSphere, servlet filters are not initialized on web application startup, but rather, on first access. This can cause problems when deploying certain apps to Liferay DXP. To configure servlet filters to initialize on application startup (i.e., deployment), you need to set the following `webcontainer` custom properties in your WebSphere application server:

```
com.ibm.ws.webcontainer.initFilterBeforeInitServlet = true
com.ibm.ws.webcontainer.invokeFilterInitAtStartup = true
```

To set `webcontainer` custom properties in the WebSphere application server, follow the instructions here in WebSphere's documentation.

Setting up JVM Parameters for Liferay DXP

Next, in the WebSphere profile you created for Liferay DXP, you must set an argument that supports Liferay DXP's Java memory requirements. You'll modify this file:

```
[Install Location]/WebSphere/AppServer/profiles/your-profile/config/cells/your-cell/nodes/your-node/servers/your-server/server.xml
```

Add `maximumHeapSize="2048"` inside the `jvmEntries` tag. For example:

```
<jvmEntries xmi:id="JavaVirtualMachine_1183122130078" ... maximumHeapSize="2048">
```

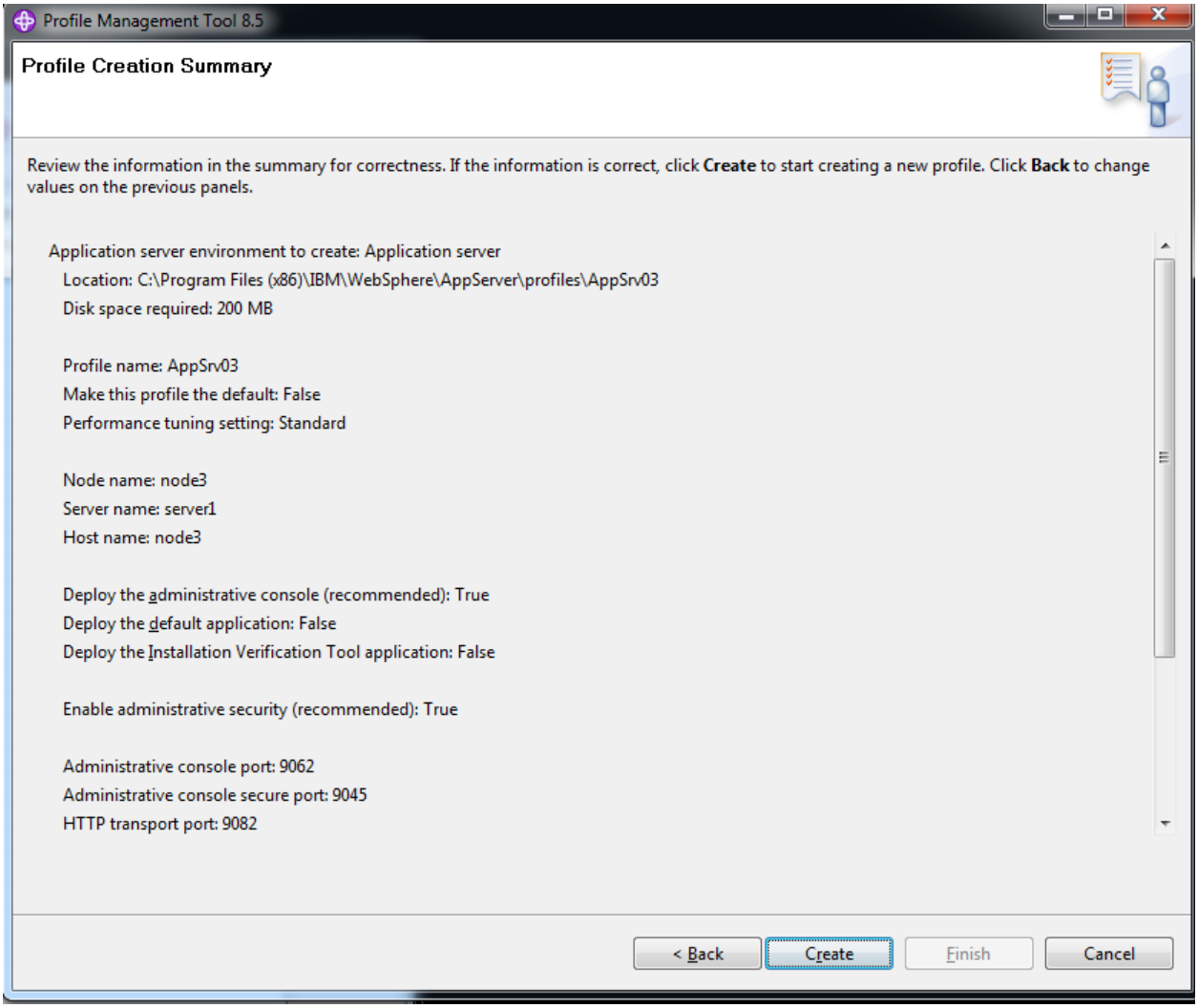


Figure 36.3: Example of the settings before creating the profile.

Note: The JVM parameters used here are defaults intended for production systems development purposes. Administrators can change the settings to other values that best address their specific environments. These will need to be tuned depending on need.

Administrators can set the UTF-8 and GMT time zone properties in the `server.xml` file. This is required or else special characters will not be parsed correctly. Add the following inside the `jvmEntries` tag:

```
<jvmEntries xmi:id="JavaVirtualMachine_1183122130078" ...genericJvmArguments="-Dfile.encoding=UTF-8 -Duser.timezone=GMT">
```

Important: For Liferay DXP to work properly, the application server JVM must use the `GMT` time zone and `UTF-8` file encoding.

Alternately, you can set the UTF-8 and GMT properties from the WebSphere Admin Console. (See below.)

Removing the secureSessionCookie Tag

In the same profile, you should delete a problematic `secureSessionCookie` tag that can cause Liferay DXP startup errors. Note that this is just a default setting; once Liferay DXP is installed, you should tune it appropriately based on your usage.

In `[Install Location]/WebSphere/AppServer/profiles/your-profile/config/cells/your-cell/cell.xml`, Delete the `secureSessionCookie` tag containing `xmi:id="SecureSessionCookie_1"`.

If this tag is not removed, an error similar to the one here may occur:

```
WSVR0501E: Error creating component com.ibm.ws.runtime.component.CompositionUnitMgrImpl@d74fa901
com.ibm.ws.exception.RuntimeWarning: com.ibm.ws.webcontainer.exception.WebAppNotLoadedException: Failed to load webapp: Failed to load webapp: SRVE8111E: Th
```

Installing Liferay DXP's Dependencies

You must now install Liferay DXP's dependencies. Recall that earlier you downloaded two ZIP files containing these dependencies. Install their contents now:

1. `liferay-dxp-digital-enterprise-dependencies-[version].zip`: Unzip this file and place its contents in your WebSphere application server's `[Install Location]/WebSphere/AppServer/lib/ext` folder. If you have a JDBC database driver JAR, copy it to this location as well.

2. `liferay-dxp-digital-enterprise-osgi-[version].zip`: Unzip this file and place its contents in the `[Liferay Home]/osgi` folder (create this folder if it doesn't exist). This is typically `[Install Location]/WebSphere/AppServer/profiles/your-profile/liferay/osgi`.

Before starting the server, verify that all the following jars have been copied to the correct folders. Optional jars are available (italics) and are used to optimize Liferay performance which must be added to this folder. Required jars in bold are from the `liferay-digital-enterprise-dependencies-7.0-ga1 zip`. The following files should be present within the `lib/ext` (WebSphere Application) folder:

1. `activation.jar`
2. `com.liferay.registry.api-1.0.4.jar`
3. `hsql.jar`
4. A JDBC database jar (e.g. MySQL, MariaDB, IBM DB2, Postgres)
5. `persistence.jar`
6. `portal-kernel.jar`
7. `portlet.jar`

The following folders should be present within the `/liferay/osgi` folder:

1. `Configs`
2. `Core`
3. `Marketplace`
4. `Target-platform`
5. `Test`

Once you've installed these dependencies, start the server profile you created for Liferay DXP. Once it starts, you're ready to configure your database.

Database Configuration

If you want WebSphere to manage the database connections, follow the instructions below. Note this is not necessary if you're planning on using Liferay DXP's standard database configuration; in that case, skip this section. You'll set your database information in Liferay DXP's setup wizard after the install.

Note: Although Liferay DXP's embedded database is fine for testing purposes, you **should not** use it for production Liferay DXP instances.

```
![ WebSphere JDBC providers](./images-dxp/websphere-jdbc-providers.png)
```

1. Start WebSphere.
2. Open the Administrative Console and log in.
3. Click *Resources → JDBC Providers*.
4. Select a scope and then click *New*.
5. Select your database type, provider type, and implementation type. If you select a predefined database, the wizard fills in the name and description fields for you. If the database you want to use isn't listed, select *User-defined* from the *Database type* field and then fill in the *Implementation Class Name*. For example, if you are using MySQL, select *Database type → User-defined*, and then enter `com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource` in *Implementation Class Name*. Click *Next* when you are finished.
6. Clear any text in the classpath settings. You already copied the necessary JARs to a location on the server's classpath. Click *Next*.
7. Review your settings and click *Finish*. The final configuration should look like this:

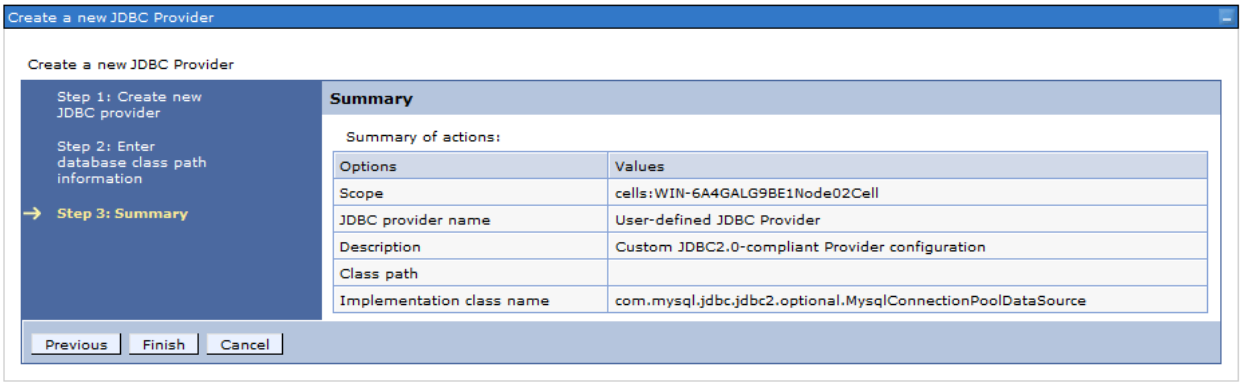


Figure 36.4: Completed JDBC provider configurations.

8. Click your new provider configuration when it appears in the table, and then click *Data Sources* under *Additional Properties*. Click *New*.
9. Enter *liferaydatabasesource* in the *Data source name* field and `jdbc/LiferayPool` in the *JNDI name* field. Click *Next*.

10. Click *Next* in the remaining screens of the wizard to accept the default values. Then review your changes and click *Finish*.

11. Click the data source when it appears in the table and then click *Custom Properties*. Now click the *Show Filter Function* button. This is the second from last of the small icons under the *New* and *Delete* buttons.

12. Type *user* into the search terms and click *Go*.

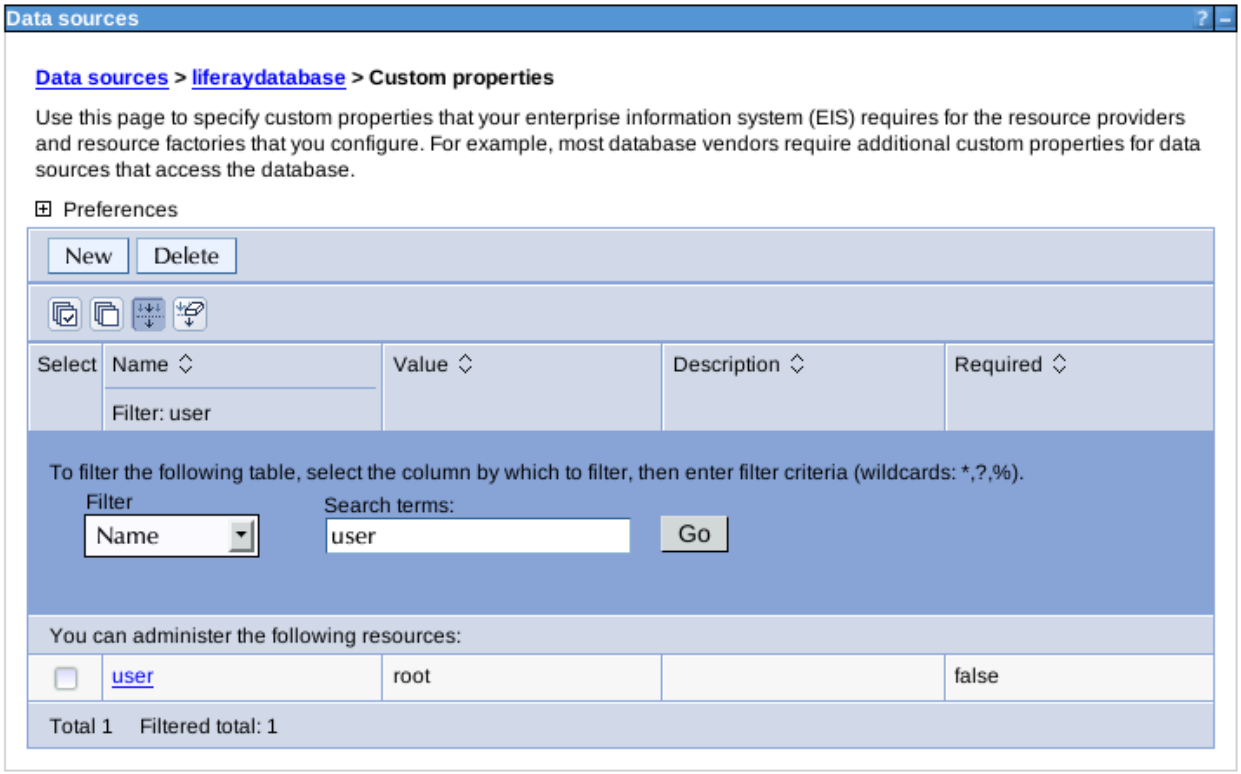


Figure 36.5: Modifying data source properties in WebSphere

13. Select the *user* property and give it the value of the user name to your database. Click *OK* and save to master configuration.

14. Do another filter search for the *url* property. Give this property a value that points to your database. For example, a MySQL URL would look like this:

    ```
    jdbc:mysql://localhost/lportal?useUnicode=true&characterEncoding=UTF-8&useFastDateParsing=false
    ```

    Click *OK* and save to master configuration.

15. Do another filter search for the *password* property. Enter the password for the user ID you added earlier as the value for this property. Click *OK* and save to master configuration.

16. Go back to the data source page by clicking it in the breadcrumb trail. Click the *Test Connection* button. It should connect successfully.

Once you've set up your database, you can set up your mail session.

Mail Configuration

If you want WebSphere to manage your mail sessions, use the following procedure. If you want to use Liferay DXP's built-in mail sessions, you can skip this section.

Creating a WebSphere-Managed Mail Session (Optional)

1. Click *Resources → Mail → Mail Providers*.
2. Click the Built-In Mail Provider for your node and server.
3. Click *Mail Sessions* and then click the *New* button.
4. Give your mail session a name of *liferaymail* and a JNDI name of `mail/MailSession`. Fill in the correct information for your mail server in the sections *Outgoing Mail Properties* and *Incoming Mail Properties*. Click *OK* and then save to the master configuration.
5. Click your mail session when it appears in the table and select *Custom Properties* under the *Additional Properties* section. Set any other JavaMail properties required by your mail server, such as the protocol, ports, whether to use SSL, and so on.
6. Click *Security → Global Security* and de-select *Use Java 2 security to restrict application access to local resources* if it is selected. Click *Apply*.

Note that you may also need to retrieve a SSL certificate from your mail server and add it to WebSphere's trust store. See WebSphere's documentation for instructions on this.

Verifying WebSphere Mail Provider

To validate that the mail session has been configured correctly, there are a number of ways to test this once the WAR has been deployed, the server has started, and the user has signed in as the system administrator. One quick way to validate is to create a new user with a valid email account. The newly created user should receive an email notification. The logs should display that the SMTP server has been pinged with the correct port number listed.

Enable Cookies for HTTP Sessions

WebSphere 8.5.5.9 restricts cookies to HTTPS sessions by default. If you're using HTTP instead, this prevents users from signing in to Liferay DXP and displays the following error in the console:

```
20:07:14,021 WARN  [WebContainer : 1][SecurityPortletContainerWrapper:341]
User 0 is not allowed to access URL http://localhost:9081/web/guest/home and
portlet com_liferay_login_web_portlet_LoginPortlet
```

This occurs because Liferay DXP can't use the HTTPS cookie when you use HTTP. The end result is that new sessions are created on each page refresh. Follow these steps to resolve this issue in WebSphere:

1. Click *Application Servers → server1 → Session Management* → Enable Cookies
2. De-select *Restrict cookies to HTTPS sessions*
3. Click *Apply*
4. Click *Save*

Enable UTF-8

If you did not add the `-Dfile.encoding=UTF-8` property in the `server.xml`, you can do so in the Administrative Console.

1. Click *Application Servers → server1 → Process definition*.
2. Click *Java Virtual Machine* under *Additional Properties*.
3. Enter `-Dfile.encoding=UTF-8` in the *Generic JVM arguments* field.
4. Click *Apply* and then *Save* to master configuration.

Once the changes have been saved, Liferay DXP can parse special characters if there is localized content.

Deploy Liferay DXP

Now you're ready to deploy Liferay DXP!

1. In WebSphere's administrative console, click *Applications → New Application → New Enterprise Application*.
2. Browse to the Liferay DXP `.war` file, select it, and click *Next*.
3. Leave *Fast Path* selected and click *Next*. Ensure that *Distribute Application* has been checked and click *Next* again.
4. Choose the WebSphere runtimes and/or clusters where you want Liferay DXP deployed. Click *Next*.
5. Select the virtual host to deploy Liferay DXP on and click *Next*.
6. Map Liferay DXP to the root context (/) and click *Next*.
7. Select the *metadata-complete attribute* setting that you want to use and click *Next*.
8. Ensure that you have made all the correct choices and click *Finish*. When Liferay DXP has installed, click *Save to Master Configuration*.

You've now installed Liferay DXP!

Setting the JDK Version for Compiling JSPs

Liferay DXP requires that its JSPs are compiled on Java 8. To ensure that WebSphere does this, shut down WebSphere after you've deployed the Liferay DXP `.war` file. Navigate to the `WEB_INF` folder and add the following setting to the `ibm-web-ext.xml` or in most cases the `ibm-web-ext.xmi` file:

```
<jsp-attribute name="jdkSourceLevel" value="18" />
```

The exact path to the `ibm-web-ext.xmi` file depends on your WebSphere installation location and Liferay DXP version, but here's an example:

```
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/localhostNode01Cell/applications/liferayXX.ear/deployments/liferayXX/liferayXX.war/WEB-INF/ibm-web-ext.xmi
```

Note that the Liferay DXP `.war` comes pre-packaged with the `ibm-web-ext.xmi` file; this format is functionally the same as `.xml` and WebSphere recognizes both formats. For more general information on how WebSphere compiles JSPs see IBM's official documentation for WebSphere Application Server 8.5.5.x.

Now restart WebSphere.

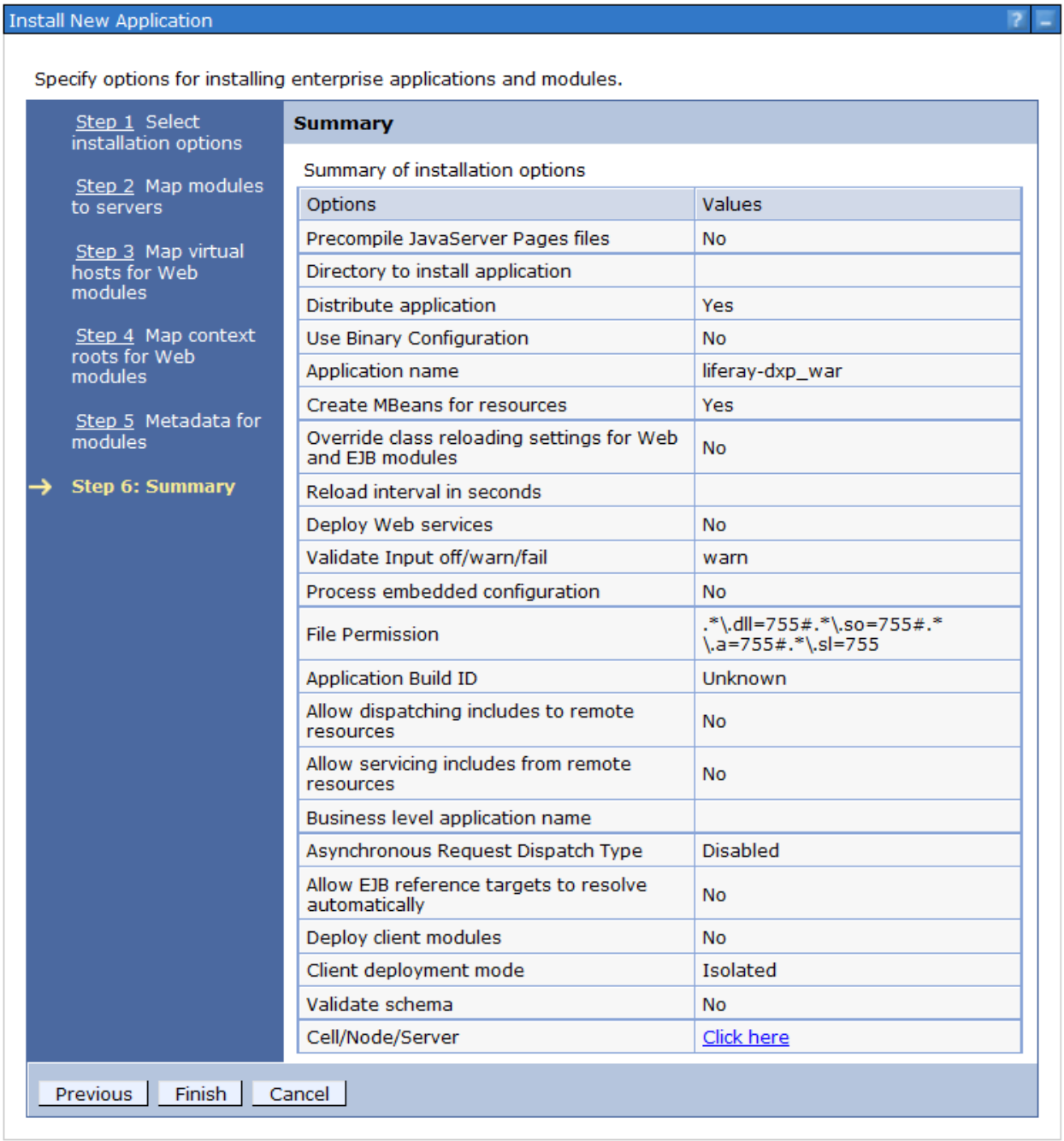


Figure 36.6: Review your deployment options before deploying.

Enabling Security for Portal Access Control Lists

When you are ready to start using apps from Liferay Marketplace, you must enable Liferay's Portal Access Control Lists (PACL) to enforce security policies on Marketplace applications. To do so, you must enable Java Security on your WebSphere server and specify a security policy to grant Liferay DXP access to your server.

In WebSphere's administrative console, go to *Security → Global Security*. Check the box to enable Java 2 security, and click *Apply*. Save to the master configuration.

Now you must configure the security policy for the IBM JRE that WebSphere runs on. With your WebSphere server shut down, open the following security policy file:

```
[WebSphere-Install-Location]/java_1.8_64/jre/lib/security/java.policy
```

Replace this file's contents with the following configuration:

```
grant {
    permission java.security.AllPermission;
};
```

This configuration opens all permissions. As best practices, opening all permissions when starting Liferay Digital Enterprise 7.0 for the first time allows for the creation of all required processes. Users can change security settings later and restrict which plugins have access. (Users have the options of using either the application server's security policies or Liferay DXP's.) See this article concerning a Known Issues when generating PACL.

Once this permission has been set, users can start Liferay DXP.

Start Liferay DXP

1. If you plan to use Liferay DXP's setup wizard, skip to the next step. If you wish to use WebSphere's data source and mail session, create a file called `portal-ext.properties` in your Liferay Home folder. Place the following configuration in the file:

   ```
   jdbc.default.jndi.name=jdbc/LiferayPool
   mail.session.jndi.name=mail/MailSession
   setup.wizard.enabled=false
   ```

2. In the WebSphere administrative console, navigate to *Enterprise Applications*, select the Liferay DXP application, and click *Start*. While Liferay DXP is starting, WebSphere displays a spinny little graphic. Don't watch it too closely, or you might get hypnotized.

3. In Liferay DXP's setup wizard, select and configure your database type. Click *Finish* when you're done. Liferay DXP then creates the tables it needs in the database.

Congratulations! You've installed Liferay DXP on WebSphere!

36.10 Installing Elasticsearch

Liferay DXP uses Elasticsearch to index its content. By default, Liferay DXP uses Elasticsearch as an embedded service. It works, but this is not a supported configuration for a production server. Feel free to use it while you're testing or developing, but when you're ready to put your site in production, you'll need to run Elasticsearch as a standalone process. This is better anyway, because it frees you to design your infrastructure the way you want it. If you've got hardware or a VM to spare, you can separate your search infrastructure

from Liferay DXP and reap some performance gains by putting search on a separate box. If you're more budget-conscious, you can still increase performance by running Elasticsearch in a separate, individually tunable JVM on the same box.

Installing Elasticsearch for Liferay DXP is pretty easy and takes only six steps:

1. Find the version of Elasticsearch that's embedded in the version of Liferay DXP you have, and then download that version from Elastic's website.
2. Install Elasticsearch by extracting its archive to the system where you want it to run.
3. Install some required Elasticsearch plugins.
4. Name your Elasticsearch cluster.
5. Configure Liferay DXP to connect to your Elasticsearch cluster.
6. Restart Liferay DXP and reindex your search and spell check indexes.

Note: Before continuing, make sure you have set the `JAVA_HOME` environment variable

If you have multiple JDKs installed, make sure Elasticsearch and Liferay DXP are using the same version. You can specify this in `[Elasticsearch Home]/bin/elasticsearch.in.sh`:

```
JAVA_HOME=/path/to/java
```

Now you'll actually perform these steps, and when you're done, you'll have a production-ready instance of Liferay DXP up and running. After you're done following the installation guide, refer to the Configuring Elasticsearch article for more details on configuring Liferay DXP for Elasticsearch. For more information on installing a search engine, see here.

Step One: Find the Right Version of Elasticsearch

If Liferay DXP isn't running, start it.

Visit port 9200 on localhost to access the embedded Elasticsearch:

```
http://localhost:9200
```

A JSON document is returned that varies slightly, but should look similar to this:

```
{
  "name" : "Wiz Kid",
  "cluster_name" : "LiferayElasticsearchCluster",
  "version" : {
    "number" : "2.4.0",
    "build_hash" : "8ff36d139e16f8720f2947ef62c8167a888992fe",
    "build_timestamp" : "2016-08-27T13:32:39Z",
    "build_snapshot" : false,
    "lucene_version" : "5.5.2"
  },
  "tagline" : "You Know, for Search"
}
```

The version of Elasticsearch that's running is the value of the `"number"` field. In this example, it's `2.4.0`.

Elasticsearch 6: Elasticsearch 6 is supported for Liferay Digital Enterprise systems running Fix Pack 79 or later: see the compatibility matrix for the exact version ranges supported for your Fix Pack level. The latest Liferay Portal, GA 7 at the time of this writing, does not support Elasticsearch 6.5. Instead use Elasticsearch 6.1. In 7.0, Elasticsearch version 2.x remains the default, embedded version. To install Elasticsearch 6.5.x,

1. Make sure you're running at least Digital Enterprise FP-79.
2. Install Elasticsearch 6.5.x (follow steps 2-4 in this article for guidance).
3. Install the *Liferay Connector to Elasticsearch 6*, version 1.1.0+, from Marketplace and stop the default Elasticsearch adapter.
4. Configure the Elasticsearch 6 connector (see step 5 below for guidance).

To learn more about upgrading an existing system to Elasticsearch 6.5, read the upgrade article.

Now that you know the version of Elasticsearch you need, go to Elastic's website and download that version.

Step Two: Install Elasticsearch

Most of this step entails deciding where you want to run Elasticsearch. Do you want to run it on the same machine as Liferay DXP, or do you want to run it on its own hardware? The answer to this question comes down to a combination of the resources you have available and the size of your installation. Regardless of what you decide, either way you get the benefit of a separately tunable search infrastructure.

Once you have a copy of the right version of Elasticsearch, extract it to a folder on the machine where you want it running. That's all there is to this step.

Step Three: Install Elasticsearch Plugins

Install the following required Elasticsearch plugins:

- `analysis-icu`
- `analysis-kuromoji`
- `analysis-smartcn`
- `analysis-stempel`

To install these plugins, navigate to Elasticsearch Home and enter

```
./bin/plugin install [plugin-name]
```

Replace *[plugin-name]* with the Elasticsearch plugin's name.

Elasticsearch 6.5: The `plugin` executable was renamed in Elasticsearch 6, to `elasticsearch-plugin`. The general command syntax for installing plugins to 6.5 is

```
./bin/elasticsearch-plugin install [plugin-name]
```

Step Four: Name Your Elasticsearch Cluster

A *cluster* in Elasticsearch is a collection of nodes (servers) identified as a cluster by a shared cluster name. The nodes work together to share data and workload. A one node cluster is discussed here; to create a multi-node cluster, please refer to Elastic's documentation.

Now that you've installed Elastic, it sits in a folder on your machine, which is referred to here as `[Elasticsearch Home]`. To name your cluster, you'll define the cluster name in both Elasticsearch and in Liferay DXP. First, define it in Elasticsearch. Edit the following file:

```
[Elasticsearch Home]/config/elasticsearch.yml
```

Uncomment the line that begins with `cluster.name`. Set the cluster name to whatever you want to name your cluster:

```
cluster.name: LiferayElasticsearchCluster
```

Of course, this isn't a very imaginative name; you may choose to name your cluster `finders_keepers` or something else you can remember more easily. Save the file.

Now you can start Elasticsearch. Run the executable for your operating system from the `[Elasticsearch Home]/bin` folder:

```
./elasticsearch
```

Elasticsearch starts, and one of its status messages includes a transport address:

```
2016-05-03 16:33:28,358][INFO ][transport] [Hobgoblin II] publish_address {127.0.0.1:9300}, bound_addresses {[::1]:9300}, {127.0.0.1:9300}
```

Take note of this address; you'll need to give it to your Liferay DXP server so it can find Elasticsearch on the network.

Step Five: Configure Liferay DXP to Connect to your Elasticsearch Cluster

---

Elasticsearch 6.5: Before continuing, install the Liferay Connector to Elasticsearch 6 application from Liferay Marketplace and stop the default Elasticsearch 2.x adapter, which connects to Elasticsearch 2.x.

1. Create a file called

   ```
   com.liferay.portal.bundle.blacklist.internal.BundleBlacklistConfiguration.config
   ```

   If you already have one of these files in `Liferay Home/osgi/configs`, just open it.

2. Place the Bundle Symbolic Name of the Elasticsearch connector in the `blacklistBundleSymbolicNames` property:

   ```
   blacklistBundleSymbolicNames=["com.liferay.portal.search.elasticsearch"]
   ```

3. Place the file in `Liferay Home/osgi/configs`.

Alternatively, deactivate the Elasticsearch connector using the App Manager in *Control Panel → Apps → App Manager*. If you're a Digital Enterprise customer, use the blacklist feature as described above. The App Manager relies on the `osgi/state` folder to "remember" the state of the bundle. If you delete this folder (recommended during patching) the Elasticsearch connector will be reinstalled and started automatically.

1. Search for *elasticsearch* in the App Manager. Find the Liferay Portal Search Elasticsearch module and click the edit (()) button. Choose the Deactivate option. This leaves the bundle installed, but stops it in the OSGi runtime.
2. Once you've downloaded the LPKG file with the Elasticsearch 6 adapter, place it in Liferay Home's `deploy` folder. Find more detailed information on deploying Marketplace applications here.

Now you're ready to configure Liferay DXP. Start Liferay DXP if you haven't already, log in, and then click on *Control Panel → Configuration → System Settings → Foundation*. Find *Elasticsearch* (or *Elasticsearch 6*)in the list of settings and click on it. Now you can configure it. Here are the options you need to change:

Cluster name: Enter the name of the cluster as you defined it in Elasticsearch.

Operation mode: Defaults to EMBEDDED. Change it to REMOTE to connect to a standalone Elasticsearch.

Transport addresses: Enter a delimited list of transport addresses for Elasticsearch nodes. Here, you'll enter the transport address from the Elasticsearch server you started. The default value is `localhost:9300`, which will work.

When finished, click *Save*. You're almost done.

Step Six: Restart Liferay DXP and Reindex

Stop and restart Liferay DXP. When it's back up, log in as an administrative user and

1. Click on *Control Panel → Configuration → Server Administration*
2. Click the *Execute* button for *Reindex all search indexes*. When you do that, you should see some messages scroll up in the Elasticsearch log.
3. Click Execute for *Reindex all spell check indexes*.

For more details refer to the Elasticsearch installation guide.

You're almost done! The only thing left is to make sure Marketplace is working and optionally configure portal security.

36.11 Liferay DXP Trial Plugin Installation

For Liferay customers who are evaluating Liferay DXP on a trial basis, **the plugins can be accessed from within the Marketplace section of the Control Panel in the product**.

At a later date, the Marketplace website on Liferay.COM will be accessible to Liferay DXP Trial customers. For now, please access EE Plugins through the portal installation itself.

The Installation Process

1) Sign into your Liferay.COM account (LRDC) on the home page. If you don't have an account, please register by clicking the same link.

2)You must be running Liferay DXP (Trial License is OK).

3) After signing in as an Admin in your Liferay DXP Trial server, go to the Control Panel and configure the Liferay.COM (LRDC) account to login to the Marketplace (click "Store"). Authorize Marketplace to access your local account.

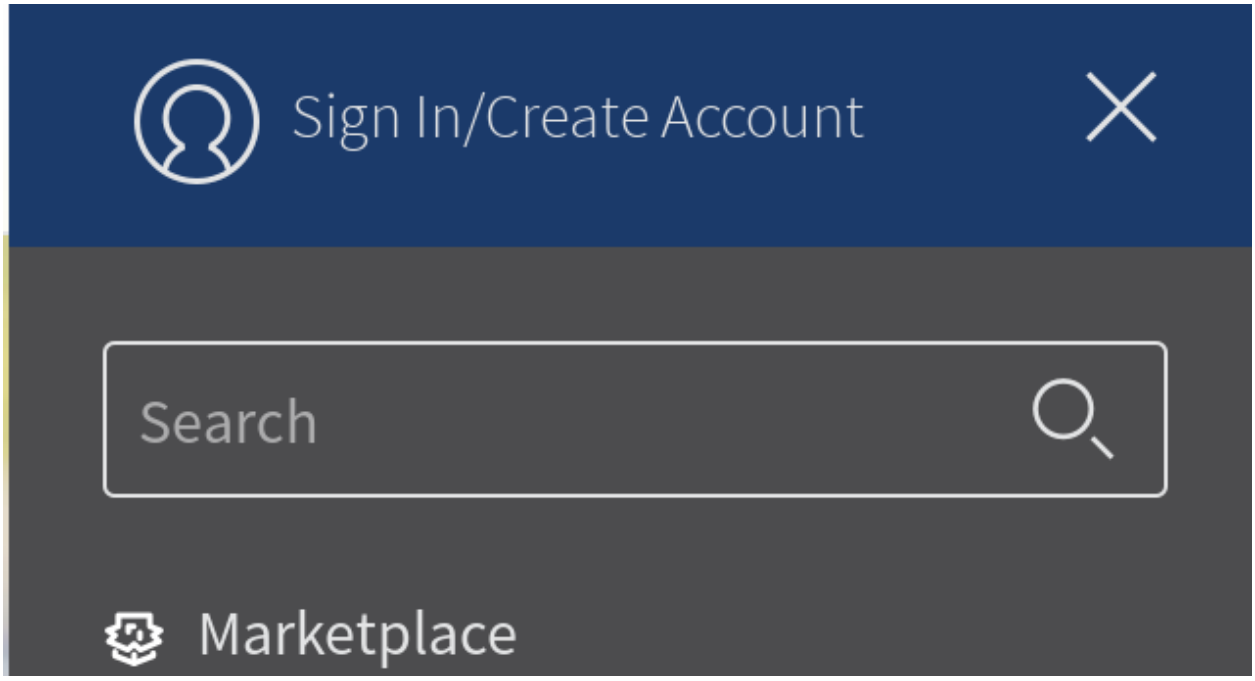


Figure 36.7: Click the hamburger menu to sign in or create an account.

4) Important: Once signed into the Store, click on the *Purchased* link, and then click on the *EE* subtab. Here you can see a list of Liferay DXP plugins that are installed, as well as options to update or install certain plugins. To install a plugin, click the *Install* button located to the far right of the respective plugin.

5) Trial users have access to the Customer Portal during their trial. This also includes access to the Liferay DXP-specific plugins from the Liferay Marketplace website on Liferay.COM (LRDC). Trial users do not have access to our customer Knowledge Base and Help Center (our support ticketing system).

Note: We are currently working on providing access to Liferay DXP Plugins on the Marketplace website for Liferay DXP Trial customers, which would also include access from within the portal installation.

FAQ

Q: Where are the "Liferay DXP Trial Plugins"?

A: There is no such thing. The Liferay DXP Plugins in Liferay Marketplace are the same ones that you get to try out with your Liferay DXP Trial license for your portal. The Liferay DXP license (Trial or official Liferay DXP Subscriber) gives you access to the Liferay DXP plugins. (Also, there is no difference code-wise or release-wise between a Liferay DXP Trial installation and a regular Liferay DXP non-Trial installation. The only difference is the license.)

Q: Why can't I go to liferay.com/marketplace? Why can't I "purchase" from the Marketplace site?

A: DXP Trial users must use the Marketplace accessed from within the product's Control Panel (instructions above). You do not need to "purchase" any DXP plugins because if you access Marketplace from within the Control Panel, Marketplace sees that you have an DXP license installed and gives access to DXP plugins. Official DXP Subscription customers (i.e., non Trial) will be able to log into liferay.com with their designated DXP subscriber login and access all DXP plugins through the Marketplace website.

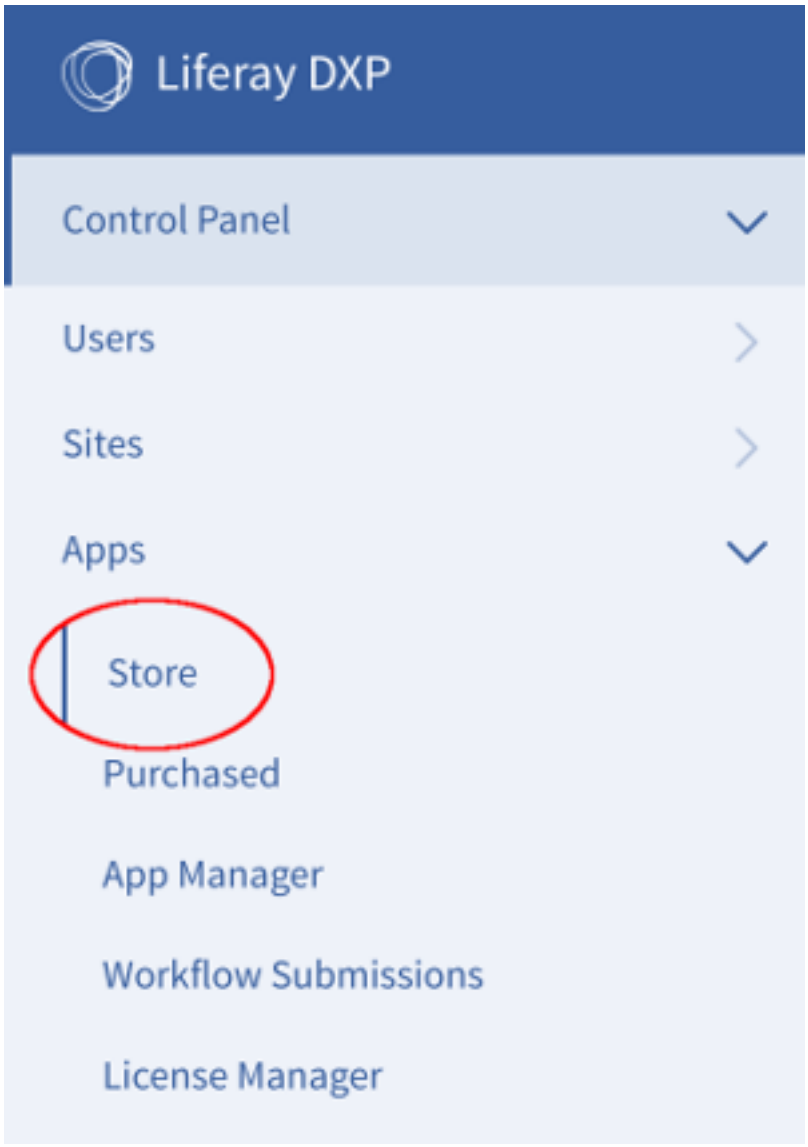


Figure 36.8: Click the *Store* link and authorize Marketplace to access your local account.

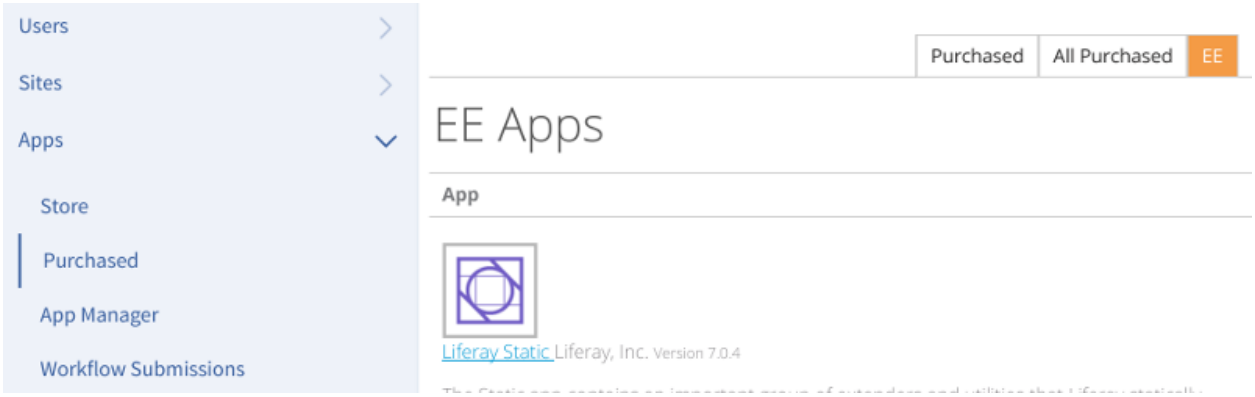


Figure 36.9: The trial plugins are available as plugins already purchased.

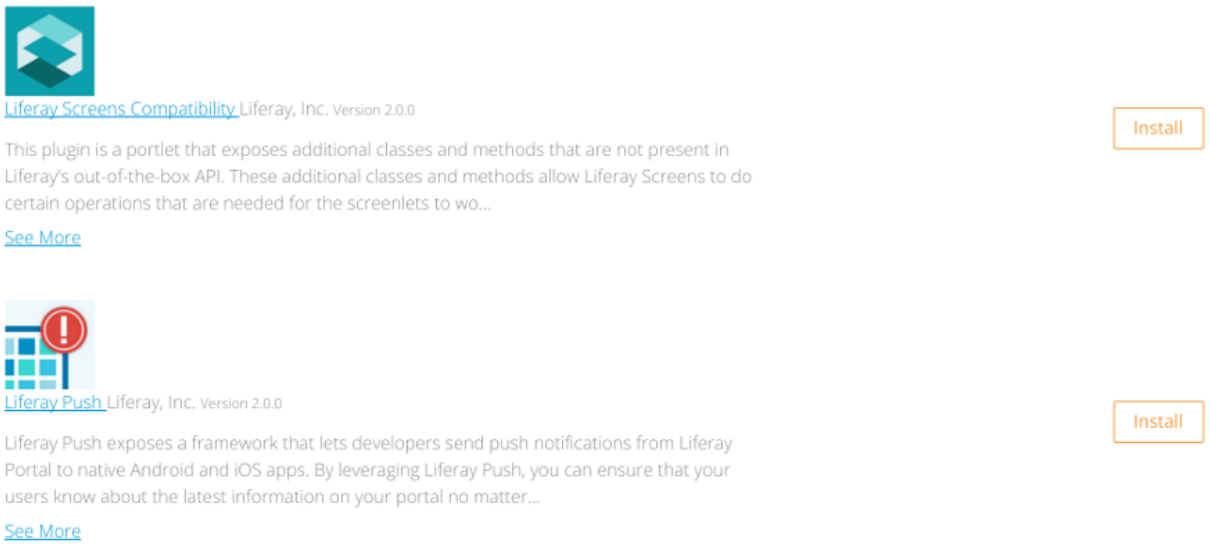


Figure 36.10: Click the *Install* button to install a plugin.

Q: Why are the plugins under the "Purchased" tab? If I click on the "DXP Marketplace" link, it does not allow me to get the DXP plugins.

A: Once you're signed into the Store, click on the *Purchased* tab, then click on the *EE* subtab.

Q: What happens when DXP Trial customers become official Liferay Digital Experience subscribers?

A: They will still be able to complete the above process, or they can also visit the Liferay Marketplace website.

Q: Do DXP Trial customers get the DXP source code?

A: No, they can only install the plugin. The DXP source code becomes available once they are official Liferay DXP Enterprise subscribers.

Q: Can this process of installing DXP plugins be used from Liferay Portal CE (Community Edition)?

A: No, the Marketplace must detect that you are running Liferay DXP.

36.12 Activating Liferay DXP

There are 2 ways to activate your Liferay DXP instance:

- With an XML activation key that you request and receive from Liferay Support.
- Online activation through Liferay Connected Services (LCS). Liferay DXP 7.0 introduced LCS as a way to activate Liferay DXP instances. LCS can also install fix packs, monitor each instance's performance, and help administrators automatically manage Liferay DXP subscriptions. See the LCS documentation for instructions on activating your instances with LCS.

Note: You must use LCS for activation of containerized instances, cloud deployments, and instances that use Liferay Analytics Cloud and/or elastic subscriptions. Otherwise, you don't have to use LCS for activation. You can instead request an XML activation key from Liferay Support.

36.13 Setting Up Marketplace and Portal Security

Liferay Marketplace is more than just a store for Liferay applications. Under the hood, it not only provides the store, it also provides Liferay DXP's application deployment features. For this reason, you must ensure that Marketplace works well on your installed system. The first thing you should do is make sure Marketplace can run and configure itself. The second thing you should do is enable Liferay's Portal Access Control List, or PACL.

First, you'll learn about some scenarios in which Marketplace fails to run, but they can all be worked around. Next, you'll configure PACL.

Server is Firewalled without Access to the Internet

Your server might be behind a firewall that prevents access to the Internet. Or your security policy might not allow direct download and installation from the Internet. In these cases, you have two options:

1. From an Internet-enabled computer, download the Marketplace plugin from https://www.liferay.com/marketplace/download. Then allow Liferay DXP to auto deploy it by dropping the downloaded `.lpkg` file into the Liferay DXP deploy folder.

2. Alternately, once you have the downloaded `.lpkg` file, use the Liferay App Manager to deploy the plugin. This option is especially helpful if the application server does not support hot deploy. See Installing Apps Manually.

Application Server Does Not Support Hot Deploy

If your application server does not support hot deploy, you can't leverage Liferay DXP's auto deploy feature. You can, however, manually deploy the plugin in two steps:

1. Use Liferay DXP's tools to pre-deploy the file.

2. Then use your app server's tools to do the actual deployment.

Limited Database Access

Some production environments do not have the necessary database permissions for Liferay DXP and its plugins to maintain their tables. In these cases:

1. Grant the Liferay DXP database user temporary full rights to the database.

2. Install Liferay DXP and start it so that it populates its database.

3. Once the database is created, remove the permissions for creating tables and dropping tables from the Liferay DXP database user.

See the Setting Up Liferay DXP's Database with Restrictive Permissions section above for more information. Note that many sophisticated Liferay DXP apps—not just the Marketplace plugin—require new tables when deployed. If your environment restricts database access, you may need to repeat the above steps whenever you deploy a new app to the Liferay DXP.

Configuring Liferay DXP Security

Liferay Marketplace is an online store for obtaining applications that run on the Liferay DXP platform. These applications are provided not only by Liferay DXP, but also by partners and independent developers who want you to install and use their applications on your server. Many of these applications are excellent and we recommend that you try them out for yourself.

However, because many of the applications on Marketplace are *not* provided by Liferay DXP, a question arises: how do you know these applications are doing what they're advertised to do? There is a vetting process that they go through before they're allowed on Marketplace, but if the source code is not provided, there's no way for even Liferay DXP to know if an app has been properly represented. For this reason, Liferay DXP implements a security feature known as the Portal Access Control List, or PACL.

PACL forces an application to declare up front the functions from Liferay DXP's APIs that it calls. Anything that's not declared is not allowed to run. It's similar to what you might see on your mobile phone when you install an app: you get to see the Liferay DXP API functions the app uses, and then you can decide if you want to install that app based on the permissions it requires. This way, you see right away what portal data that app can access and the app can do nothing else: you're protected–if you have PACL enabled. So if you plan to use apps downloaded from Marketplace, it's important to make sure PACL is enabled.

By default, Liferay DXP's bundles have PACL turned off. The reason for this is that there is a small performance penalty for having PACL enabled. Since the only reason to have PACL enabled is to install untrusted third party apps from Marketplace (and not everybody does that), we decided to leave PACL turned off by default. This way, your portal performs as fast as possible.

The bottom is line that if you intend to use Marketplace apps, you should enable PACL. We provide manually installation documentation for all the app servers supported by Liferay DXP. Each of those sections has a section that explains how to enable Java security for that app server, which is a prerequisite for enabling PACL. Once you have Java security enabled, PACL can be enabled by adding one line to your `portal-ext.properties` or `portal-setup-wizard.properties` file:

```
portal.security.manager.strategy=liferay
```

Save the file. If Liferay DXP is running, restart it. Your portal is now configured to check PACL-enabled Marketplace apps against their declared permissions.

Please note that if you installed Liferay DXP manually, there may be further configuration you need to do to enable PACL. Please check the relevant installation instructions for your app server for further information.

Congratulations! Liferay DXP is now installed and ready for production.

36.14 Choosing IPv4 or IPv6

Liferay DXP supports both the IPv4 and IPv6 address formats. By default, Liferay DXP uses IPv4 addresses. If you're on an IPv6 network, you'll need to change the configuration. If you'd like more information on the basics of these protocols, you can check out the reason for using IPv6 addresses, and its technical details.

To configure your portal to validate IPv6 addresses, you must complete a few simple steps:

1. Assuming you're using the Tomcat app server for your portal, edit the `setenv.sh` or `setenv.bat` file in the `${TOMCAT_HOME}/bin` folder and set `-Djava.net.preferIPv4Stack=false` in `CATALINA_OPTS`.
2. Create a `portal-ext.properties` file in your portal's Liferay Home folder (if one does not already exist) and set the `tunnel.servlet.hosts.allowed` property to the target hosts you want to allow (e.g., *0:0:0:0:0:0:0:1*).

Chapter 37

Configuring Liferay DXP

Once you have Liferay DXP installed, it's time to configure it to the specifics of your environment. This means doing things like setting the time zone and language, selecting your document repository type, setting up security, tuning and more. These topics and more are discussed here.

37.1 Locales and Encoding Configuration

Liferay DXP lets you display content based on language, time zone, "right to left" (that is, languages such as Hebrew, Arabic, and Persian), and lets you localize user names and titles. Administrators can localize specific core UI messages so that the messages display in certain languages.

Time Zones

Time zones can be set in the Control Panel and theoretically in the JVM (but this must be set to GMT: see below).

To configure the time zone and customize the default language in the Control Panel, administrators can make changes at the Instance level.

1. Navigate to the *Control Panel* → *Configuration*.
2. Click *Instance Settings*.
3. Click on the *Miscellaneous* tab.

Using the central left and right arrows, administrators can add or remove available languages and locales. You can also modify these properties in your `portal-ext.properties` file. You'll find them in the Company section:

```
company.default.locale=en_GB
```

As an example, the above property changes the locale to English, Great Britain.

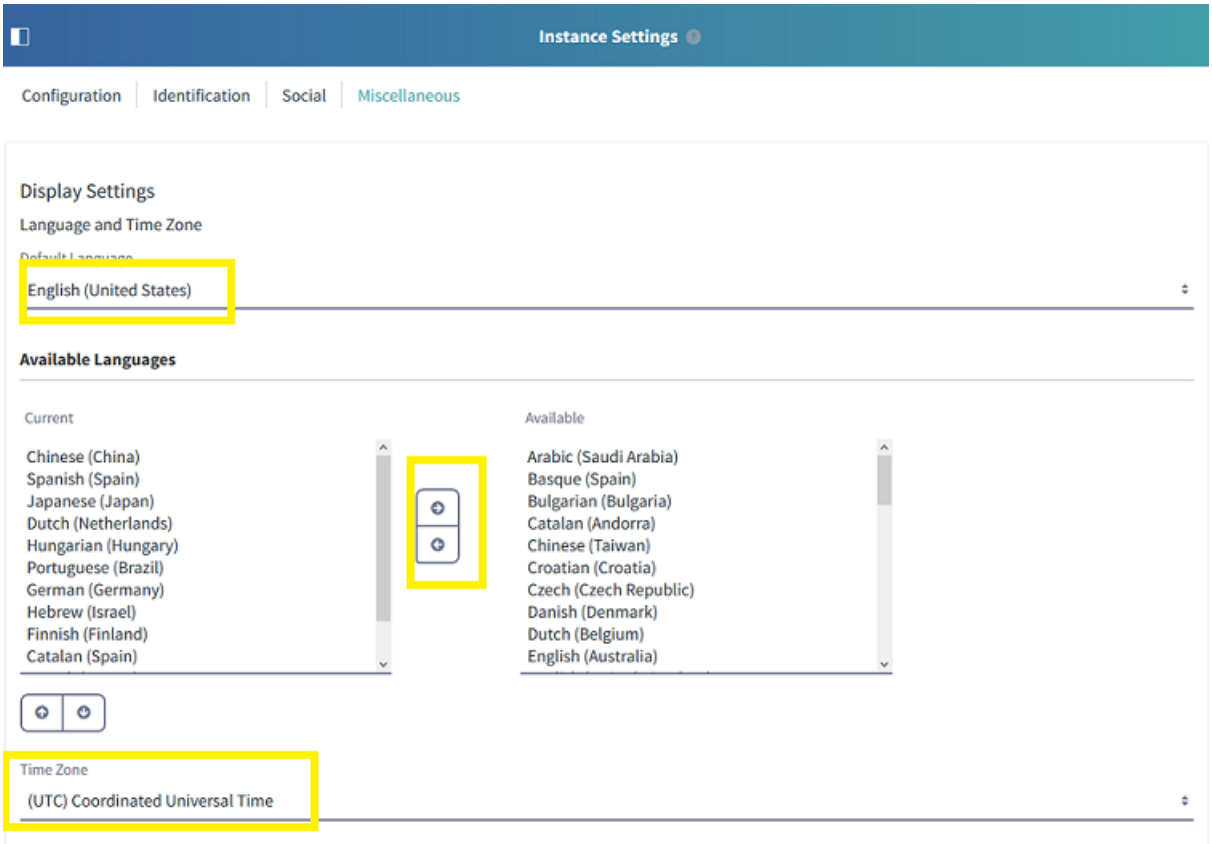


Figure 37.1: You can change the default and available languages and the time zone in Instance Settings.

Set the JVM Time Zone to GMT

It is possible to set time zones at the JVM level. However, users will encounter issues such as Calendar Events and Web Content articles displaying the wrong dates. This happens because the system assumes each date stored in the database is stored in GMT time. When the system needs to display one stored date to the end users, Liferay DXP calculates the display date based on the current date of the *application server*. This date is affected by the configured JVM level time zone and the stored GMT format date. In order to make sure the display date is calculated correctly, the time zone must be configured to GMT at the JVM level. Otherwise, it will result in incorrect time zone offset and cause the display date to be wrongly calculated and displayed.

Friendly URLs and Locales

In addition to configuring Liferay DXP's instance settings, users can also provide unique URLs for specific languages using the `I18nServlet` by editing Liferay DXP's `web.xml` file:

```
<servlet-mapping>
    <servlet-name>I18n Servlet</servlet-name>
    <url-pattern>/ar/*</url-pattern>
</servlet-mapping>
.
.
.
<servlet-mapping>
    <servlet-name>I18n Servlet</servlet-name>
    <url-pattern>/de/*</url-pattern>
</servlet-mapping>
```

The defaults that Liferay provides should be sufficient for nearly all circumstances. Because this requires stopping and possibly redeploying Liferay DXP (depending on your app server), test the defaults and make sure you really need to modify these settings. If you're clustered, you must make these changes on all nodes.

Modifying Language Keys

Developers can add or modify certain core UI messages (e.g. *Your request completed successfully.*) by modifying the language keys that ship by default.

Right to Left

For languages that are displayed right to left, modify the `language.properties` using the following properties:

```
lang.dir=rtl
lang.line.begin=right
lang.line.end=left
```

Localizing User Names

Users can change the prefix and suffix values for a locale. For example, for Spanish, the `language_es.properties` file contains these values:

```
lang.user.name.field.names=prefix,first-name,last-name
lang.user.name.prefix.values=Sr,Sra,Sta,Dr,Dra
lang.user.name.required.field.names=last-name
```

For more information, see Using Liferay Language Settings.

Related Topics

Using Liferay Language Settings
    Modifying Liferay DXP's Language Keys
    Overriding a Module's Language Keys

37.2 Document Repository Configuration

There are several options available for configuring how Liferay DXP's Documents and Media library stores files. Each option is a *store* which can be configured through the `portal-ext.properties` file by setting the `dl.store.impl=` property.

By default, Liferay DXP uses a document library store option called Simple File Store to store documents and media files on a file system (local or mounted). The store's default root folder is `[Liferay Home]/data/document_library`. You can specify a different root directory from within System Settings. To access System Settings, open the *Menu* () and navigate to *Control Panel* → *Configuration* → *System Settings*. From System Settings, navigate to *Foundation* and then search for and select the entry *Simple File System Store*. For the store's *Root dir* value, specify a path relative to Liferay Home or an absolute path; then click the *Update* button. The document library store switches immediately to the new folder.

You can also use an entirely different method for storing documents and media files:

Simple File System Store: uses the file system (local or a mounted share) to store files.

Advanced File System Store: in addition to using the file system (local or a mounted share) to store files, Advanced File System Store nests the files into more directories by version, for faster performance and to store more files. separate from Liferay DXP to store files.

DBStore (Database Storage): stores files in the Liferay DXP database.

S3Store (Amazon Simple Storage): uses Amazon's cloud-based storage solution.

CMIS Store (deprecated as of Liferay DXP Digital Enterprise 7.0 Fix Pack 14 (SP3) and Liferay Portal CE 7.0 GA4): uses a system

JCRStore (deprecated as of Liferay DXP Digital Enterprise 7.0 Fix Pack 14 (SP3) and Liferay Portal CE 7.0 GA4): stores files to a JSR-170 compliant document repository. You can use any JCR client to access the files. The files are stored to the server's file system by default. You can optionally configure JCRStore to store files in a database.

Here are the details for each one.

Using the File System Store

This is the default store. It's a simple file storage implementation that uses a local folder to store files. You can use the file system for your clustered configuration, but you'd have to make sure the folder to which you point the store can handle things like concurrent requests and file locking. For this reason, you need to use a Storage Area Network or a clustered file system.

The file system store was the first store created for Liferay DXP and is heavily bound to the Liferay DXP database. By default, documents are stored in a `document_library` subfolder of the `data` folder in a Liferay DXP bundle. Of course, you can change this path to anything you want in System Settings.

This store creates a folder structure based on primary keys in the Liferay DXP database. If, for example, you upload a presentation with the file name `workflow.odp` into a folder called *stuff*, the file system store creates a folder structure that looks like the figure below.

The folder path used by Liferay DXP for storing documents is this:

```
/companyId/folderId/numericFileEntryName/versionNumber
```

The first folder name is the company ID to which the site belongs. The second folder name is the Documents and Media folder's ID where the document resides. The third folder name is the document's numeric file entry name. Finally, the fourth name is a version number used for storing multiple versions of the document.

Note: A document's numeric file entry name is distinct from the document ID; be careful not to confuse the two! Each has an independent counter. The numeric file entry name is used in the folder path for storing the document but the document ID is not. The numeric file entry name can be found in the `name` column of the `DLFileEntry` table in Liferay DXP's database; the document ID can be found in the `fileEntryId` column of the same table.

As you can see, the File System Store binds your documents very closely to Liferay DXP, and may not be exactly what you want. If you've been using the default settings for a while and need to migrate your documents, Liferay DXP provides a migration utility in the Control Panel in *Server Administration → Data Migration*. Using this utility, you can move your documents very easily from one store implementation to another.

Using the Advanced File System Store

Liferay DXP's advanced file system store is similar to the default file system store. Like that store, it saves files to the local file system–which, of course, could be a remote file system mount. It uses a slightly different folder structure to store files, which is pictured below.

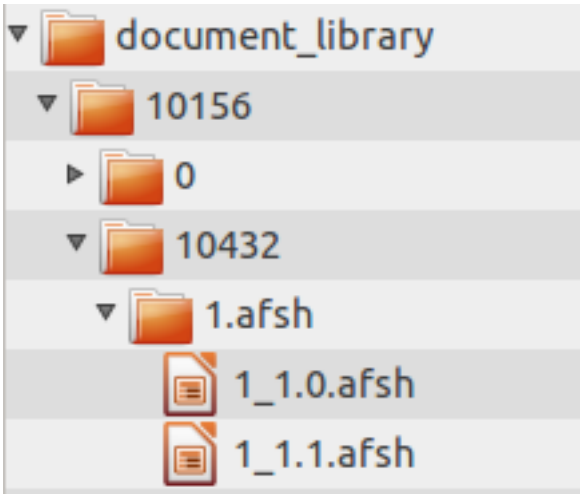


Figure 37.2: The advanced file system store creates a more nested folder structure than the file system store.

So what makes the advanced file system store *advanced*? Several operating systems have limitations on the number of files which can be stored in a particular folder. The advanced file system store overcomes this limitation by programmatically creating a structure that can expand to millions of files, by alphabetically nesting the files in folders. This not only allows for more files to be stored, but also improves performance as there are fewer files stored per folder.

The same rules apply to the advanced file system store as apply to the default file system store. To cluster this, you must point the store to a network mounted file system that all the nodes can access, and that networked file system must support concurrent requests and file locking. Otherwise, you may experience data corruption issues if two users attempt to write to the same file at the same time from two different nodes.

Note: Since Liferay DXP 7.0 Fix Pack 14 and Liferay Portal 7.0 CE GA4, to configure the advanced file system store you must configure both the `portal-ext.properties` and `.config` files.

Create the following file inside your app server's `osgi/configs` folder:

```
com.liferay.portal.store.file.system.configuration.AdvancedFileSystemStoreConfiguration.cfg
```

Note: you can also generate the config file by exporting the configurations from the Control Panel under *System Settings → Foundation → Advanced File System Store*.

Configure the property shown below:

Property | Default | Required `rootDir` | `data/document_library` | `true`

You may need to update the `rootDir` property to correctly reflect where your document library is stored.

Next, configure `portal-ext.properties`:

```
dl.store.impl=com.liferay.portal.store.file.system.AdvancedFileSystemStore
```

With both the `.config` file and `portal-ext.properties` configured, you can start Liferay DXP.

You may decide the advanced file system store for whatever reason doesn't serve your needs. If this is the case, you can of course mount other file systems into the documents and media library. In addition to this,

you can also redefine the Liferay DXP store to use one of three other supported protocols. We'll look at these next.

Using Amazon Simple Storage Service

Amazon's simple storage service (S3) is a cloud-based storage solution that you can use with Liferay DXP. All you need is an account, and you can store your documents to the cloud from all nodes, seamlessly.

When you sign up for the service, Amazon assigns you unique keys that link you to your account. In Amazon's interface, you can create "buckets" of data optimized by region. Once you've created these to your specifications, use these instructions to connect your S3 account to Liferay DXP.

If you are using Tomcat as your app server, it doesn't contain a `SAXParser`. You must include this property in `system-ext.properties`:

```
org.xml.sax.driver=com.sun.org.apache.xerces.internal.parsers.SAXParser
```

Other app servers also need this configuration if they don't contain a `SAXParser`. Remember to place your `system-ext.properties` file in a folder that resides in your Liferay DXP installation's class path (e.g., `/WEB-INF/classes/`).

Note: No action is required to support AWS Signature Version 4 request authorization.

Consult the Amazon Simple Storage documentation for additional details on using Amazon's service.

Using the CMIS Store

Though you can mount as many different CMIS (Content Management Interoperability Services) repositories as you like in the Documents and Media library, you may wish also to redefine the Liferay DXP repository to point to a CMIS repository as well. Why? Users might want to create a folder or upload content to the Liferay DXP repository. It would be nice if that Liferay DXP repository was connected to a clustered CMIS repository by the administrator without having to mount it through the UI. The CMIS store allows you to do just that.

Note: CMIS Store is not suitable for production use and is deprecated as of Liferay DXP Digital Enterprise 7.0 Fix Pack 14 (SP3) and Liferay Portal CE 7.0 GA4. Because it can have performance issues with large repositories, it's recommended that you use one of the other configuration repositories listed above, such as Advanced File System Store, to store your Documents and Media files. This deprecation does not affect the use of external repositories. You can still connect to external repositories using CMIS.

If you wish to use the CMIS store, follow the instructions here to set it up. The Liferay DXP repository is connected to CMIS via the CMIS store. As long as all nodes are pointing to your CMIS repository, everything in your Liferay DXP cluster should be fine, as the CMIS protocol prevents multiple simultaneous file access from causing data corruption.

Using the JCR Store

Liferay DXP supports as a store the Java Content Repository standard. Under the hood, Liferay DXP uses Jackrabbit, a project from Apache, as its JSR-170 compliant document repository. By default, Jackrabbit is configured to store the documents on the local file system where Liferay DXP is installed, in the `[Liferay Home]/liferay/jackrabbit` folder. Inside this folder is Jackrabbit's configuration file, called `repository.xml`.

Note: JCR Store is deprecated as of Liferay DXP Digital Enterprise 7.0 Fix Pack 14 (SP3) and Liferay Portal CE 7.0 GA4. You should use one of the other documentation repositories listed above.

Using the default settings, the JCR store is not very different from the file system stores, except you can use any JCR client to access the files. You can, however, modify Jackrabbit's configuration so it stores files in a database that can be accessed by all nodes, and so that it operates as a cluster within Liferay DXP's cluster.

To move the default repository location to a shared folder, you do not need to edit Jackrabbit's configuration file. Instead, follow the instructions here. Change it to point to a shared folder that all the nodes can see. A new Jackrabbit configuration file is then generated in that location, and you'll have to edit that file to modify Jackrabbit's configuration.

Note that because of file locking issues, this isn't the best way to share Jackrabbit resources, unless you're using a networked file system that can handle concurrency and file locking. If you have two people logged in at the same time uploading content, you could encounter data corruption using this method, and because of this, we don't recommend it for a production system. Instead, if you want to use the Java Content Repository in a cluster, you should redirect Jackrabbit into your database of choice. You can use the Liferay DXP database or another database for this purpose. This requires editing Jackrabbit's configuration file.

The default Jackrabbit configuration file has sections commented out for moving the Jackrabbit configuration into the database. This has been done to make it as easy as possible to enable this configuration. To move the Jackrabbit configuration into the database, comment out the sections relating to the file system and comment in the sections relating to the database. These by default are configured for a MySQL database. If you are using another database, you will need to modify the configuration, as there are changes to the configuration file that are necessary for specific databases. For example, the default configuration uses Jackrabbit's `DbFileSystem` class to mimic a file system in the database. While this works well in MySQL, it doesn't work for all databases. For example, an Oracle database requires the `OracleFileSystem`.

Modify the JDBC database URLs so they point to your database. This, of course, must be done on all nodes of the cluster. Don't forget to create the database first and grant the user ID you are specifying in the configuration file access to create, modify, and drop tables. After this, be sure to uncomment the `<Cluster/>` section at the bottom of the file. For further information, it's best to check out the Jackrabbit documentation.

Once you've configured Jackrabbit to store its repository in a database, the next time you bring up Liferay DXP, the necessary database tables are created automatically. Jackrabbit, however, does not create indexes on these tables, and so over time this can be a performance penalty. To fix this, you must manually go into your database and index the primary key columns for all the Jackrabbit tables.

Note that this configuration doesn't perform as well as the advanced file system store, because you're storing documents in a database instead of on the file system. But it does have the benefit of clustering well. For example, you can store documents and media files in your Liferay DXP instance's database using DBStore. To enable DBStore, add the following `dl.store.impl` portal property to a `portal-ext.properties` file in your Liferay Home:

```
dl.store.impl=com.liferay.portal.store.db.DBStore
```

Remember to restart your Liferay DXP server after updating your `portal-ext.properties` file in order for your customizations to take effect.

Configuration Details

There are properties related to document library stores that have been moved from `portal-ext.properties` to OSGi configuration files. The following mapping shows you how to configure those properties if needed:

File Store

From portal-ext.properties: dl.store.impl=com.liferay.portal.store.file.system.FileSystemStore
To osgi/configs/com.liferay.portal.store.file.system.configuration.FileSystemStoreConfiguration.cfg:
Property | Default | Required rootDir | data/document_library | false

Advanced File Store

Since Liferay DXP 7.0 Fix Pack 14 and Liferay Portal 7.0 CE GA4, both the portal-ext.properties and .config files are required to configure the advanced file system store.

From portal-ext.properties: dl.store.impl=com.liferay.portal.store.file.system.AdvancedFileSystemStore
To osgi/configs/com.liferay.portal.store.file.system.configuration.AdvancedFileSystemStoreConfiguration.cfg:
Property | Default | Required rootDir | data/document_library | true

S3

From portal-ext.properties: dl.store.impl=com.liferay.portal.store.s3.S3Store
To osgi/configs/com.liferay.portal.store.s3.configuration.S3StoreConfiguration.cfg:
Property | Default | Required accessKey | | false secretKey | | false s3Region | us-east-1 | false bucketName | | true s3StorageClass | STANDARD | false httpClientMaxConnections | 50 | false cacheDirCleanUpExpunge | 7 | false cacheDirCleanUpFrequency | 100 | false

---

Note: Amazon S3 requires a SAXParser from the application server to operate. For some application servers (e.g. Tomcat), it's necessary to define a SAXParser in order to prevent errors while utilizing this store. This may be set in system-ext.properties. For example,

```
org.xml.sax.driver=com.sun.org.apache.xerces.internal.parsers.SAXParser
```

Warning: If a database transaction rollback occurs in a Document Library that uses a file system based store, file system changes that have occurred since the start of the transaction won't be reversed. Inconsistencies between Document Library files and those in the file system store can occur and may require manual synchronization. All stores except DBStore are vulnerable to this limitation.

CMIS Store

From portal-ext.properties: dl.store.impl=com.liferay.portal.store.cmis.CMISStore
To osgi/configs/com.liferay.portal.store.cmis.configuration.CMISStoreConfiguration.cfg:
Property | Default | Required repositoryUrl | http://localhost:8080/alfresco/service/api/cmis | true credentialsUsername | none | true credentialsPassword | none | true systemRootDir | Liferay Home | true

JCR

From portal-ext.properties: dl.store.impl=com.liferay.portal.store.jcr.JCRStore
To osgi/configs/com.liferay.portal.store.jcr.configuration.JCRStoreConfiguration.cfg:
Property | Default | Required initializeOnStartup | false| true wrapSession | true | true moveVersionLabels | false | true workspaceName | liferay | true nodeDocumentlibrary | documentlibrary | true jackrabbitRepositoryRoot | data/jackrabbit | true jackrabbitConfigFilePath | repository.xml | true jackrabbitRepositoryHome | home | true jackrabbitCredentialsUsername | none | true jackrabbitCredentialsPassword | none | true

---

Note: Use the absolute path for `jackrabbitRepositoryRoot`, `jackrabbitConfigFilePath`, and `jackrabbitRepositoryHome`.

Please refer to the Document Library property reference for a complete list of supported customizations. You can customize features such as the maximum allowed size of documents and media files, the list of allowed file extensions, which types of files should be indexed, and more.

37.3 Liferay DXP Clustering

Liferay DXP can serve everything from the smallest to the largest web sites. Out of the box, it's configured optimally for a single server environment. If one server isn't sufficient to serve the high traffic needs of your site, Liferay DXP scales to the size you need.

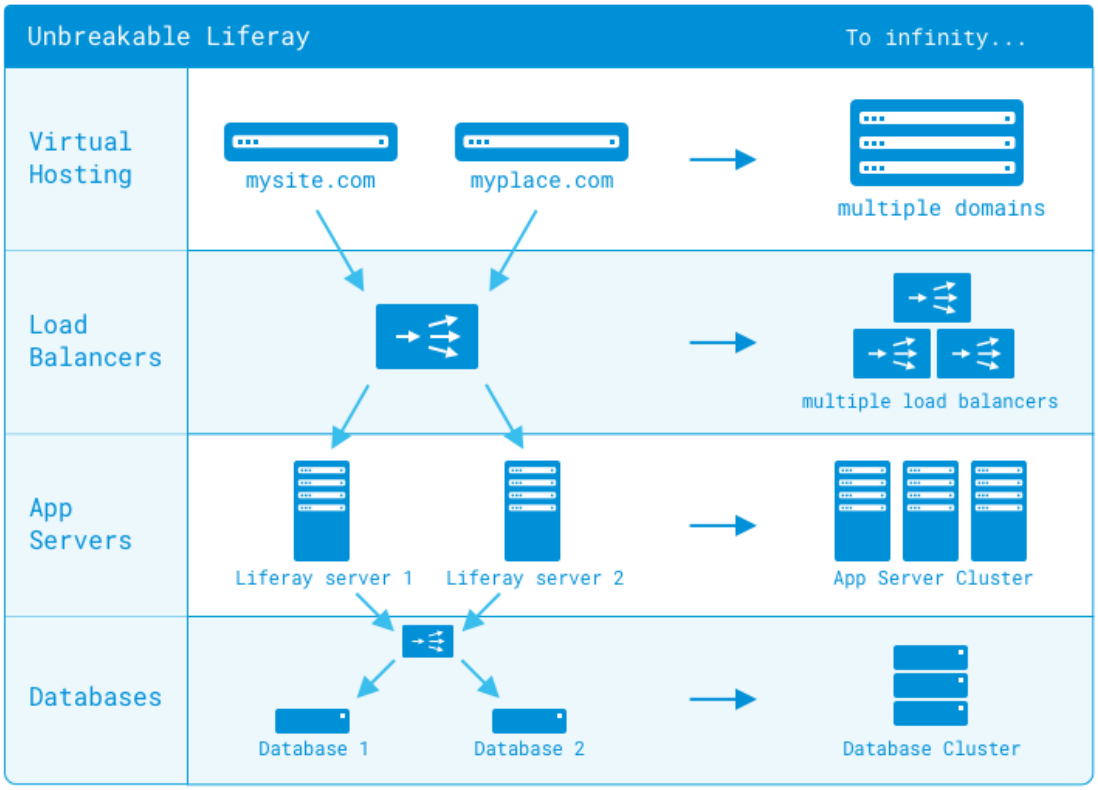


Figure 37.3: Liferay DXP is designed to scale to as large an installation as you need.

Liferay DXP works well in clusters of multiple machines (horizontal cluster) or in clusters of multiple VMs on a single machine (vertical cluster), or any mixture of the two. Once you have Liferay DXP installed in more than one application server node, there are several optimizations that need to be made. At a minimum, Liferay DXP should be configured in the following way for a clustered environment:

1. All nodes should be pointing to the same Liferay DXP database or database cluster.

2. Documents and Media repositories must have the same configuration and be accessible to all nodes of the cluster.
3. Search should be on a separate search server that is optionally clustered.
4. Cluster Link must be enabled so the cache replicates across all nodes of the cluster.
5. Hot deploy applications to each node individually.

If you haven't configured your application server to use farms for deployment, the hot deploy folder should be a separate folder for all the nodes, and plugins will have to be deployed to all of the nodes individually. This can be done via a script. If you do have farms configured, you can deploy normally to any node's deploy folder, and your farm configuration should take care of syncing the deployment to all nodes.

Many of these configuration changes can be made by adding or modifying properties in your `portal-ext.properties` file. Remember that this file overrides the defaults in the `portal.properties` file. You can also browse its contents here. It's a best practice to copy the relevant section you want to modify from `portal.properties` into your `portal-ext.properties` file, and then modify the values there.

Note: This article documents a Liferay DXP-specific cluster configuration without getting into specific implementations of third party software, such as Java EE application servers, HTTP servers, and load balancers. Please consult your documentation for those components of your cluster to configure those components. Before creating a Liferay DXP cluster, make sure your OS is not defining the hostname of your box to the local network at 127.0.0.1.

Each step defined above is covered below to give you a step by step process for creating your cluster.

1. All Nodes Should Point to the Same Liferay DXP Database

Each node should have a data source that points to one Liferay DXP database (or a database cluster) that all the nodes will share. This means, of course, Liferay DXP cannot (and should not) use the embedded HSQL database that is shipped with the bundles (but you already knew that, right?). And, of course, the database server should be on a separate system from the server which is running Liferay DXP.

You can also use a read-writer database configuration to optimize your database configuration.

Read-Writer Database Configuration

Liferay DXP allows you to use two different data sources for reading and writing. This enables you to split your database infrastructure into two sets: one optimized for reading and one optimized for writing. Since all Liferay DXP's supported databases support replication, you can use your database vendor's replication mechanism to keep the database nodes in sync.

Enabling a read-writer database is simple. In your `portal-ext.properties` file:

1. Set the default database connection pool provider to `dbcp`, `tomcat`, or `c3po`. Note, provider HikariCP does not support read/write splitting. Here's an example setting:

   ```
   jdbc.default.liferay.pool.provider=dbcp
   ```

   All the portal JDBC configuration properties are documented here.

2. Configure two different data sources for Liferay DXP to use, one for reading, and one for writing:

```
jdbc.read.driverClassName=com.mysql.jdbc.Driver
jdbc.read.url=jdbc:mysql://dbread.com/lportal?useUnicode=true&characterEncoding=UTF-8&useFastDateParsing=false
jdbc.read.username=**your user name**
jdbc.read.password=**your password**

jdbc.write.driverClassName=com.mysql.jdbc.Driver
jdbc.write.url=jdbc:mysql://dbreadwrite.com/lportal?useUnicode=true&characterEncoding=UTF-8&useFastDateParsing=false
jdbc.write.username=**your user name**
jdbc.write.password=**your password**
```

To use the JNDI instead of the JDBC data sources, set the `*.username` and `*.password` properties above to your JNDI user name and password and set these additional properties:

```
jdbc.read.jndi.name=**your read JNDI name**
jdbc.write.jndi.name=**your read-write JNDI name**
```

3. Avoid using the `default` data source, by setting this:

```
counter.jdbc.prefix=jdbc.write.
```

And if you're using a `dbcp` or `tomcat` database connection pool provider, set these:

```
jdbc.default.validationQuery=
jdbc.read.validationQuery=SELECT releaseId FROM Release_
jdbc.write.validationQuery=SELECT releaseId FROM Release_
```

These settings are related to issue LPS-64624.

4. Enable the read-writer database configuration by uncommenting the following Spring configuration files from the `spring.configs` and `spring.infrastructure.configs` properties:

```
spring.configs=\
    [..]
    META-INF/dynamic-data-source-spring.xml,\
    [..]

spring.infrastructure.configs=\
    [..]
    META-INF/dynamic-data-source-infrastructure-spring.xml,\
    [..]
```

The Spring configuration portal properties are documented here.

The next time you start Liferay DXP, it uses the two data sources you have defined. Be sure you have correctly set up your two databases for replication before starting Liferay DXP.

2. Documents and Media Library Clustering

Liferay DXP's Documents and Media Library can mount several repositories at a time while presenting a unified interface to the user. By default, users can use the Liferay DXP repository, which is already mounted. This repository is built into Liferay DXP and can use one of several different store implementations as its back-end. In addition to this, users can mount many different kinds of third party repositories. In a cluster, Documents and Media must have the exact same configuration on all nodes. If you have a separate repository you've mounted, all nodes of the cluster must point to this repository. Your avenue for improving performance at this point is to cluster your third party repository, using the documentation for the repository you have chosen. If you don't have a third party repository, you can configure the Liferay DXP repository to perform well in a clustered configuration.

The main thing to keep in mind is you need to make sure that every node of the cluster has the same access to the file store as every other node. For this reason, you must look at your store configuration.

Note that the file systems used by the `File System` or `Advanced File System` stores must support concurrent requests and file locking.

Checkpoint: To test if the sharing works well, execute the following steps:

1. On Node 1 upload a document to the Documents and Media.
2. On Node 2 download the document. The download should be successful.
3. Repeat the test with reversed roles.

---

3. Clustering Search

Search should always run on a separate environment from your Liferay DXP server. Liferay DXP supports Elasticsearch or Solr, and either of those environments can also be clustered.

For more information on how to cluster Elasticsearch, see Elasticsearch's distributed cluster setup.

Once Liferay DXP servers have been properly configured as a cluster and the same for Elasticsearch, change Liferay DXP from *embedded* mode to *remote* mode. On the first connection, the two sets of clustered servers communicate with each other the list of all IP addresses; in case of a node going down, the proper failover protocols will enable. Queries and indices can continue to be sent for all nodes.

For more information on how to cluster Solr, see Apache Solr Cloud documentation.

Once Liferay DXP servers have been properly configured as a cluster, deploy the Liferay Solr 5 Adapter on all nodes. (This app is available for download from Liferay Marketplace here.) Create a Solr Cloud (cluster) managed by *Apache Solr Zookeeper*. Connect the Liferay DXP cluster to Zookeeper and finish the final configurations to connect the two clusters.

4. Enable Cluster Link

Enabling Cluster Link automatically activates distributed caching. Distributed caching enables some RMI (Remote Method Invocation) cache listeners that are designed to replicate the cache across a cluster. Cluster Link uses Ehcache, which has robust distributed caching support. The cache is distributed across multiple Liferay DXP nodes running concurrently. The Ehcache global settings are in the `portal.properties` file.

By default Liferay does not copy cached entities between nodes. If an entity is deleted or changed, for example, Cluster Link sends an *remove* message to the other nodes to invalidate this entity in their local cache. Requesting that entity on another node results in a cache *miss*; the entity is then retrieved from the database and put into the local cache. Entities added to one node's local cache are not copied to local caches of the

other nodes. An attempt to retrieve a new entity on a node which doesn't have that entity cached results in a cache *miss*. The miss triggers the node to retrieve the entity from the database and store it in its local cache.

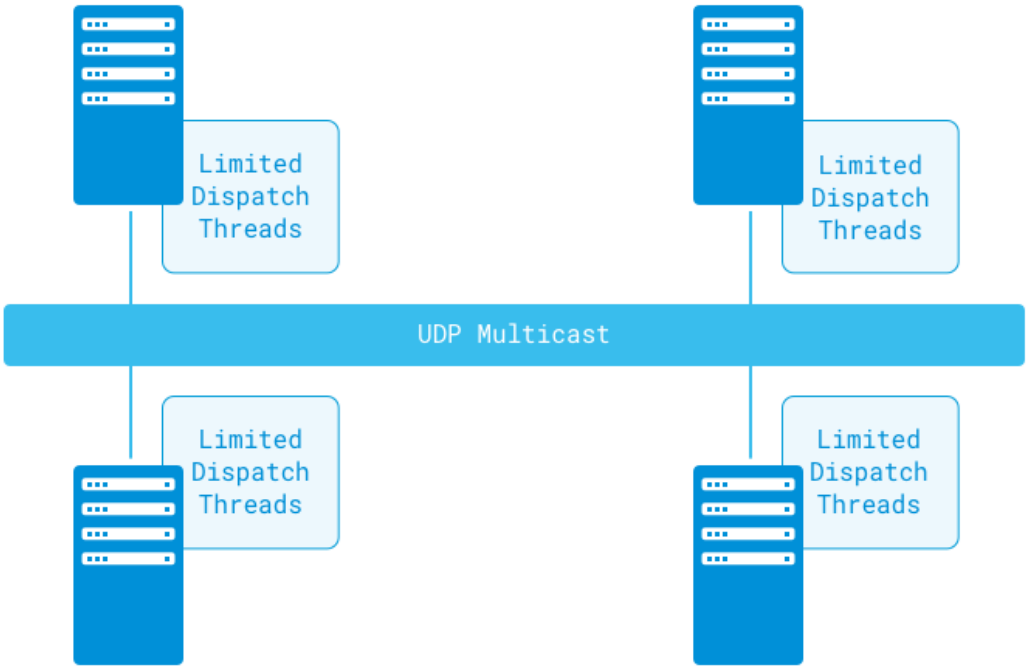


Figure 37.4: Liferay DXP's cache algorithm is extremely efficient.

To enable Cluster Link, add this property to `portal-ext.properties`:

```
cluster.link.enabled=true
```

Cluster Link depends on JGroups, and provides an API for nodes to communicate. It can

- Send messages to all nodes in a cluster
- Send messages to a specific node
- Invoke methods and retrieve values from all, some, or specific nodes
- Detect membership and notify when nodes join or leave

When you start @portal@ in a cluster, a log file message shows your cluster's name (e.g., `cluster=liferay-channel-control`):

```
-------------------------------------------------------------------
GMS: address=oz-52865, cluster=liferay-channel-control, physical address=192.168.1.10:50643
-------------------------------------------------------------------
```

Cluster Link contains an enhanced algorithm that provides one-to-many type communication between the nodes. This is implemented by default with JGroups's UDP multicast, but unicast and TCP are also available.

Multicast over UDP

When you enable Cluster Link, Liferay DXP's default clustering configuration is enabled. This configuration defines IP multicast over UDP. Liferay DXP uses two groups of channels from JGroups to implement this: a control group and a transport group. If you want to customize the channel properties, you can do so in `portal-ext.properties`:

```
cluster.link.channel.name.control=[your control channel name]
cluster.link.channel.properties.control=[your control channel properties]
```

Please see JGroups's documentation for channel properties. The default configuration sets many properties whose settings are discussed there.

Multicast broadcasts to all devices on the network. Clustered environments on the same network communicate with each other by default. Messages and information (e.g., scheduled tasks) sent between them can lead to unintended consequences. Isolate such cluster environments by either separating them logically or physically on the network, or by configuring each cluster's `portal-ext.properties` to use different sets of multicast group address and port values.

JGroups sets a bind address automatically, using `localhost` by default. In some configurations, however, `localhost` is bound to the internal loopback network (`127.0.0.1` or `::1`), rather than the host's real address. As long as DXP's `cluster.link.autodetect.address` Portal Property points to a server that's contactable, DXP uses that server to automatically detect your host's real address. Here's the default setting:

```
cluster.link.autodetect.address=www.google.com:80
```

Contacting Google may not work if your server is behind a firewall.

An alternative to detecting the host address automatically for the bind address, you can set the bind address manually in your `portal-ext.properties` file.

1. Disable address auto-detection by setting the `cluster.link.autodetect.address` property to an empty value:

   ```
   cluster.link.autodetect.address=
   ```

2. Set the following properties to your host's IP address:

   ```
   cluster.link.bind.addr["cluster-link-control"]=[place your IP address or host name here]
   cluster.link.bind.addr["cluster-link-udp"]=[place your IP address or host name here]
   ```

Your network configuration may preclude the use of multicast over TCP, so below are some other ways you can get your cluster communicating. Note that these methods are all provided by JGroups.

Checkpoint:

1. If you are binding the IP address instead of using `localhost`, make sure the right IP addresses are declared using:

   `cluster.link.bind.addr["cluster-link-control"]=localhost`
   `cluster.link.bind.addr["cluster-link-udp"]=localhost`

2. Test your load and then optimize your settings if necessary.

Unicast over TCP

If your network configuration or the sheer distance between nodes prevents you from using UDP Multicast clustering, you can configure Liferay DXP to use TCP Unicast. You'll definitely need this if you have a firewall separating any of your nodes or if your nodes are in different geographical locations.

1. Add a parameter to your app server's JVM:

   ```
   -Djgroups.bind_addr=[place your IP address or host name here]
   ```

   Use the node's IP address or host name.

2. Now you have to determine the discovery protocol the nodes should use to find each other. You have four choices:

   ```
   - TCPPing
   - JDBCPing
   - S3_Ping
   - Rackspace_Ping
   ```

   If you aren't sure which one to choose, use TCPPing. This is used in the rest of these steps; the others are covered below.

3. Download the latest `com.liferay.portal.cluster.multiple-[version].jar` file from Liferay's Nexus repository. In this JAR's `lib` folder is a file called `jgroups-[version].Final.jar`. Open it and find `tcp.xml`. Extract this file to a location accessible to Liferay DXP. Use this file on all your nodes.

4. If you're vertically clustering (i.e., you have multiple Liferay DXP servers running on the same physical or virtual system), you must change the port on which discovery communicates for all nodes other than the first one, to avoid TCP port collision. To do this, modify the TCP tag's `bind_port` parameter:

   ```
   <TCP bind_port="[some unused port]"
       ...
   />
   ```

   Since the default port is `7800`, provide some other unused port.

5. Add to the same tag the parameter `singleton_name="liferay_cluster"`. This merges the transport and control channels to reduce the number of thread pools. See JGroups documentation for further information.

   Usually, no further JGroups configuration is required. However, in a very specific case, if *(and only if)* cluster nodes are deployed across multiple networks, then the parameter `external_addr` must be set on each host to the external (public IP) address of the firewall. This kind of configuration is usually only necessary when nodes are geographically separated. By setting this, clustered nodes that are deployed to separate networks (e.g. separated by different firewalls) can communicate together. This configuration will likely be flagged in security audits of your system. See JGroups documentation for more information.

6. Save the file. Modify that node's `portal-ext.properties` file to point to it:

   ```
   cluster.link.channel.properties.control=[CONFIG_FILE_PATH]/tcp.xml
   cluster.link.channel.properties.transport.0=[CONFIG_FILE_PATH]/tcp.xml
   ```

You're now set up for Unicast over TCP clustering! Repeat this process for each node you want to add to the cluster.

JDBC Ping

Rather than use TCP Ping to discover cluster members, you can use a central database accessible by all the nodes to help them find each other. Cluster members write their own and read the other members' addresses from this database. To enable this configuration, replace the `TCPPING` tag with the corresponding `JDBCPING` tag:

```
<JDBC_PING
    connection_url="jdbc:mysql://[DATABASE_IP]/[DATABASE_NAME]?useUnicode=true&amp;characterEncoding=UTF-8&amp;useFastDateParsing=false"
    connection_username="[DATABASE_USER]"
    connection_password="[DATABASE_PASSWORD]"
    connection_driver="com.mysql.jdbc.Driver"/>
```

The above example uses MySQL as the database. For further information about JDBC Ping, please see the JGroups Documentation.

S3 Ping

Amazon S3 Ping can be used for servers running on Amazon's EC2 cloud service. Each node uploads a small file to an S3 bucket, and all the other nodes read the files from this bucket to discover the other nodes. When a node leaves, its file is deleted.

To configure S3 Ping, replace the `TCPPING` tag with the corresponding `S3_PING` tag:

```
<S3_PING
    secret_access_key="[SECRETKEY]"
    access_key="[ACCESSKEY]"
    location="ControlBucket"/>
```

Supply your Amazon keys as values for the parameters above. For further information about S3 Ping, please see the JGroups Documentation.

Other Pings

JGroups supplies other means for cluster members to discover each other, including Rackspace Ping, BPing, File Ping, and others. Please see the JGroups Documentation for information about these discovery methods.

Modifying the Cache Configuration with a Module

It's recommended to test your system under a load that best simulates the kind of traffic your system needs to handle. If you'll be serving up a lot of message board messages, your script should reflect that. If web content is the core of your site, your script should reflect that too.

As a result of a load test, you may find that the default distributed cache settings aren't optimized for your site. In this case, you should tweak the settings yourself. You can modify the Liferay DXP installation directly or you can use a module to do it. Either way, the settings you change are the same. A benefit of working with modules is that you can install a module on each node and change the settings without taking down the cluster. Modifying the Ehcache settings with a module is recommended over modifying the Ehcache settings directly.

We've made this as easy as possible by creating the project for you. Download the project and unzip it into a Liferay Workspace, in the workspace's `modules` folder. To override your cache settings, you only have to modify one Ehcache configuration file, which you'll find inside the `src/main/java` folder structure:

- resources

- ehcache
  - override-liferay-multi-vm-clustered.xml

In the sample project, this file contains a configuration for Liferay DXP's `GroupImpl` object which handles sites. You may wish to add other objects to the cache; in fact, the default file caches many other objects. For example, if you have a vibrant community, a large portion of your traffic may be directed at the message boards portlet, as in the example above. To cache the threads on the message boards, configure a block with the `MBMessageImpl` class:

```
<cache
    eternal="false"
    maxElementsInMemory="10000"
    name="com.liferay.portlet.messageboards.model.impl.MBMessageImpl"
    overflowToDisk="false"
    timeToIdleSeconds="600"
>
</cache>
```

If you're overriding these properties, it's because you want to customize the configuration for your own site. A good way to start with this is to extract Liferay's cluster configuration file and then customize it. You'll find it in the Liferay Foundation application suite's `com.liferay.portal.ehcache-[version].jar` file. You can get this JAR from the `Liferay Foundation.lpkg` file in the `osgi/marketplace` folder. The file you want is `liferay-multi-vm-clustered.xml`, in the `/ehcache` folder inside the `com.liferay.portal.ehcache-[version].jar` file. Once you have the file, replace the contents of the `override-liferay-multi-vm-clustered.xml` file above with the contents of this file. Now you'll be using the default configuration as a starting point.

Once you've made your changes to the cache, save the file, build, and deploy the module, and your settings override the default settings. In this way, you can tweak your cache settings so that your cache performs optimally for the type of traffic generated by your site. You don't have restart your server to change the cache settings. This is a great benefit, but beware: since Ehcache doesn't allow for changes to cache settings while the cache is alive, reconfiguring a cache while the server is running flushes the cache.

5. Hot Deploy to All Nodes

If you want to deploy any module or WAR file onto the cluster, it must be deployed to all nodes of the cluster. Because Liferay DXP now installs applications as OSGi bundles, this means you cannot rely on your application server's means of installing WAR files (even if you only intend to install WAR files) to deploy an application to the entire cluster. Instead, the application must be placed in Liferay DXP's `deploy` folder on each node.

This, as you might imagine, can be done with a script. Write a shell script that uploads applications to each node using sftp or some other service. This way, when you deploy an application, it is uploaded to each node's `deploy` folder and installed by each running Liferay DXP installation.

Summary

Setting up Liferay DXP on a cluster takes five steps:

1. Point all nodes at the same database or database cluster.
2. Make sure the Documents and Media repository is accessible to all nodes.
3. Install Elasticsearch or Solr on a separate system or cluster.

4. Enable Cluster Link for cache replication.
5. Hot deploy applications to each node individually.

CHAPTER 38

UPDATING A CLUSTER

Maintaining a cluster is a big responsibility. It includes deploying new and updated modules and plugins, applying fixes and improvements, making configuration changes, and more. Maximizing server uptime and minimizing risks take priority when applying changes. Liferay DXP supports standard cluster maintenance techniques, such as rolling restarts.

You should use rolling restarts if possible as it maximizes uptime and mitigates risks of changes harming your deployment. Other techniques include

- Cluster code changes.
- Non-revertible fix packs.
- Module/plugin data changes (modifying data in existing columns).
- Module/plugin data schema changes that break compatibility with the existing version. Breaking changes include but are not limited to dropping columns, changing column types, and changing data formats used in columns (such as changing from XML to JSON).
- Updating a data schema to a version outside of a Service Builder service module's required data schema range.

Since eligible changes should be done with rolling restarts, it's explained first.

38.1 Using Rolling Restarts

A rolling restart is shutting down and updating nodes one at a time (while the other nodes are running) until they're all updated. This keeps your site running while you update your cluster, whether it's physical, container, or image based.

Here's how a rolling restart works:

1. Shut down one cluster node (JVM instance).
2. Update/modify the deployment for that node (see the maintenance scenarios that follow).
3. Start the node.
4. Repeat these steps for all other cluster nodes.

User experience can be inconsistent during a rolling restart. For example, UI changes in a plugin update are only visible on the updated nodes. Users on nodes that haven't been updated see the old interface. Maintenance scenarios might have specific cases that cannot be performed in rolling restarts. The scenario descriptions mention these cases.

The maintenance scenarios eligible for rolling restarts are described below.

New Modules and Plugins

For a new plugin or module (one that does not already exist in the cluster) to be eligible for rolling restart it must not modify data, delete, or rename database columns in a way that breaks compatibility with existing plugins or modules.

Updating Existing Modules and Plugins

For a new version of an existing plugin or module to be eligible for rolling restart, it must not modify data or delete or rename database columns in a way that breaks compatibility with the existing version of the plugin or module.

Applying Fix Packs (DXP only)

The Customer Portal identifies fix packs that are not revertible, and therefore ineligible for rolling restart. All other fix packs are eligible.

Reverting Fix Packs (DXP only)

Revertible fix packs can be removed in rolling restarts.

Portal Properties controlled by `portal-ext.properties`

Portal Properties file changes can be applied in rolling restarts.

System Settings controlled by Configuration Admin Files

System configuration files can be applied in rolling restarts.

Application Server or JVM setting modifications

Modifications to application server and JVM settings can be done in rolling restarts.

Java Version Updates

Minor version updates of Java can be applied in rolling restarts. Major version updates are not supported in rolling restarts, and should instead be done when all cluster nodes are shut down.

All rolling restart-eligible updates can be applied using the rolling restart steps listed above. Other updates must be done differently as described next.

Related Topics

Liferay DXP Clustering
    Maintaining Liferay DXP
    Implementing Data Upgrades

38.2 Other Cluster Update Techniques

Several update scenarios cannot be done by rolling restart because they affect cluster communication, break compatibility with existing plugins/modules, or break Service Builder services. Also non-revertible updates are disqualified because reversing their effects requires restoring data from a backup.

Maintenance changes ineligible for rolling restart are typically done on all nodes at once, when they're shut down. The following sections describe techniques for applying these changes.

- Custom Plugin/Module Data Schema Changes
- Custom Plugin/Module Data Changes
- Non-revertible Fix Packs (DXP only)
- Service Builder Service Schema Version Changes
- Cluster Code Changes

Data schema changes are explained first.

Custom Module/Plugin Data Schema Changes

Custom module/plugin data schema changes that break compatibility with existing modules and plugins must be introduced over several releases in which the data is transitioned and maintained in old and new columns until the old column is no longer needed.

Custom Module/Plugin Data Changes

Data changes to modules or plugins you've developed require these steps:

1. Create a new column.
2. Copy the data to the new column.
3. Maintain both columns until the old column is no longer used by any cluster nodes.
4. Delete the column in the next release.

Non-revertible Fix Packs (DXP only)

The Customer Portal identifies fix packs that are not revertible. Non-revertible fix packs must be applied to nodes when they are all shut down.

Service Builder Service Schema Version Changes

A module's `Liferay-Require-SchemaVersion` (specified in its `bnd.bnd`) must match the module's schema version value in the `Release_` table. Installing a new module that has a new schema version updates the `Release_` table with that schema version and triggers a data upgrade process. If you install such a module on one node, the schema version in the `Release_` table no longer matches the `Liferay-Require-SchemaVersion` of the modules on the other nodes. The module's Service Builder services become unavailable. Such changes cannot be reverted: the database must be restored from a backup. These schema version changes must be applied while all nodes are shut down.

Cluster Code Changes

Cluster communication must stay intact. For this reason, cluster code must not be updated in rolling restarts. The Customer Portal identifies DXP fix packs that contain such changes as non-revertible. Here are packages you must not change in rolling restarts:

- `com.liferay.portal.kernel.cluster`
- `com.liferay.portal.kernel.cluster.*`
- `com.liferay.portal.kernel.exception.NoSuchClusterGroupException`
- `com.liferay.portal.kernel.scheduler.multiple`
- `com.liferay.portal.kernel.scheduler.multiple.*`
- `com.liferay.portal.cache.multiple`
- `com.liferay.portal.cache.multiple.*`
- `com.liferay.portal.scheduler.multiple`
- `com.liferay.portal.scheduler.multiple.*`

Now you know how to update your cluster using ways other than rolling restart.

Related Topics

Liferay DXP Clustering
    Maintaining Liferay DXP
    Implementing Data Upgrades

38.3 Configuring Remote Staging in a Clustered Environment

If you're running Liferay DXP as a clustered environment and you want to use remote staging, you must configure it properly for a seamless experience. In this tutorial, you'll learn how to set up remote staging in an example clustered environment scenario. The example environment assumes you have

- A Staging instance with database configurations and a file repository different from the cluster nodes
- A balancer responsible for managing the traffic flow between the cluster's nodes
- Two nodes that call two Liferay app servers (e.g., *App Server 1* and *App Server 2*), both of which are connected to the same database

The steps below also assume your web tier, application tier, and cluster environment are already configured. You may need to adjust the configurations in this tutorial to work with your specific configuration.

Let's begin!

1. You must secure the communication made between your nodes and Staging server. Add the following property to both app servers' and Staging server's `portal-ext.properties` file:

   ```
   tunneling.servlet.shared.secret=[secret]
   ```

   This secret key denies other portals access to your configured portal servers. If you'd like to set your secret key using hexadecimal encoding, also set the following property in your `portal-ext.properties` files:

   ```
   tunneling.servlet.shared.secret.hex=true
   ```

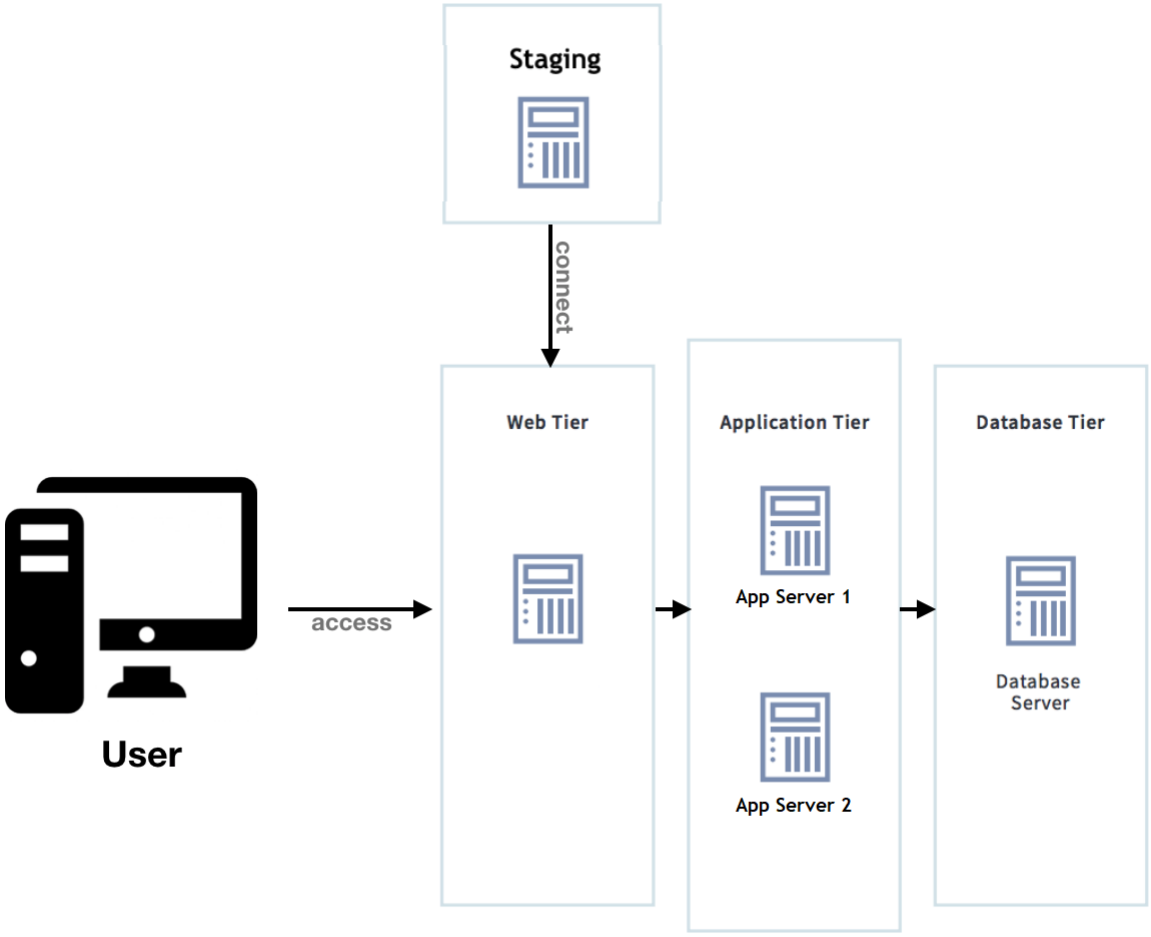


Figure 38.1: This is the assumed setup for your clustered environment.

```
**Note:** The following key lengths are supported by the available
encryption algorithms:

- *AES:* 128, 192, and 256-bit keys
- *Blowfish:* 32-448 bit keys
- *DESede (Triple DES):* 56, 112, or 168 bit keys (Liferay places an
  artificial limit on the minimum key length and does not support the 56-bit
  key length)

For example, you can use [OpenSSL](https://www.openssl.org/) to generate a
128-bit AES key:

    openssl enc -aes-128-cbc -k abc123 -P -md sha1
```

---

2. You must allow the connection between the configured IPs of your app servers and the Staging server. Open both of your app servers' `portal-ext.properties` files and add the following properties:

```
tunnel.servlet.hosts.allowed=127.0.0.1,SERVER_IP,[STAGING_IP]
tunnel.servlet.https.required=false
```

The `[STAGING_IP]` variable must be replaced by the staging server's IP addresses. The `SERVER_IP` constant can remain set for this property; it's automatically replaced by the Liferay server's IP addresses.

3. If you're validating IPv6 addresses, you must configure the app server's JVM to not force the usage of IPv4 addresses. For example, if you're using Tomcat, add the following attribute in the `$TOMCAT_HOME\bin\setenv.[bat|sh]` file.

   ```
   -Djava.net.preferIPv4Stack=false
   ```

4. Restart both app servers for the new properties to take effect.

5. Configure the *TunnelAuthVerifier* property for your nodes' app servers. There are two ways to do this:

   - **Use a `.config` file (recommended):** In the `$LIFERAY_HOME/osgi/configs` folder of one of your node Liferay DXP instances, create (if necessary) a `com.liferay.portal.security.auth.verifier.tunnel.module.configu` `default.config` file and insert the properties listed below. Creating one `.config` file configures all cluster nodes the same way. For more information on `.config` files, see the Understanding System Configuration Files article.

     ```
     enabled=true
     hostsAllowed=127.0.0.1,SERVER_IP,STAGING_IP
     serviceAccessPolicyName=SYSTEM_USER_PASSWORD
     urlsIncludes=/api/liferay/do
     ```

   - **Via System Settings:** Navigate to the *Control Panel* → *Configuration* → *System Settings* → *Foundation* → *Tunnel Auth Verifiers*. Click on the */api/liferay/do* configuration entry and add the Staging IP address to the *Hosts allowed* field. If you choose to configure the *TunnelAuthVerifier* this way, you **must** do this for all nodes (e.g., App Server 1 and App Server 2).

6. On your Staging instance, navigate to the Site Administration portion of the Product Menu and select *Publishing* → *Staging*. Then select *Remote Live*.

7. For the Remote Host/IP field, insert the balancer's IP of your web tier. Configuring the Staging instance with the balancer's IP ensures the availability of the environment at the time of publication from staging to live.

8. Enter the port on which the balancer is running into the Remote Port field.

9. Insert the remote site ID of your app servers into the Remote Site ID field. The site ID of all your app servers are the same since they are configured for the same database and are shared between nodes.

   Navigate to the Site Administration portion of the Product Menu and select *Site Settings* to find the site ID.

10. Save the Remote Live settings.

That's it! You've configured remote staging in your clustered environment.

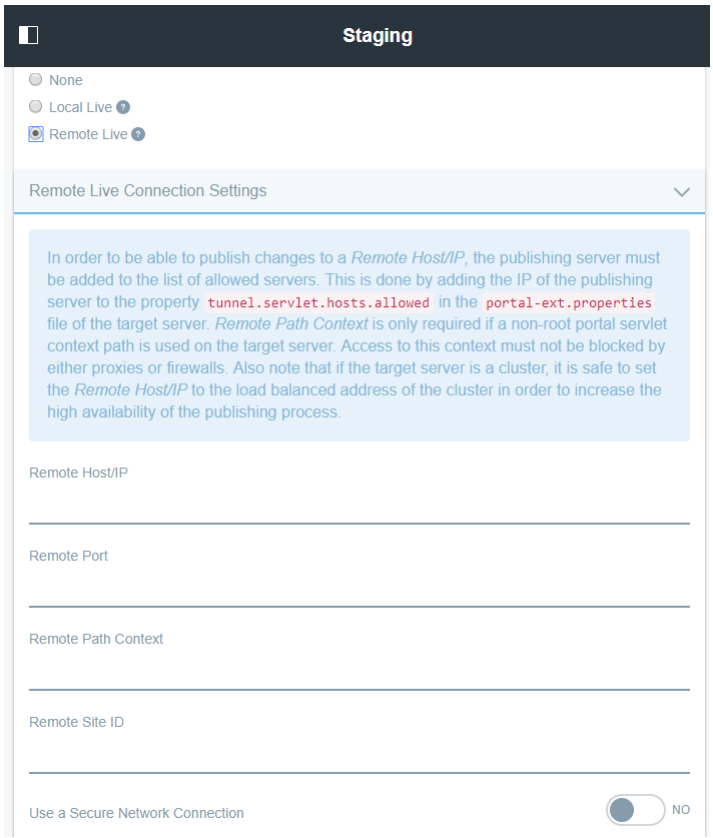


Figure 38.2: When selecting the Remote Staging radio button, you're given a list of options to configure.

CHAPTER 39

LIFERAY DIGITAL ENTERPRISE CONFIGURATION AND TUNING GUIDELINES

When tuning Liferay DXP installation, there are several factors to take into consideration; some are specific to Liferay DXP, while others are concepts that apply to all Java and Java enterprise applications. The following guidelines are meant to serve as an initial baseline from which to tune your specific deployment.

39.1 Application Server Tuning

Although the actual setting names may differ, these concepts are applicable across most application servers. For brevity, we will use Tomcat as an example. For other application servers, consult the application server provider's documentation for additional specific settings.

Database Connection Pool

The database connection pool is usually sized at roughly 30-40% of the thread pool size. The connection pool provides a connection whenever Liferay DXP needs to retrieve data from the database (e.g. user login, etc.). If this size is too small, requests queue in the server waiting for database connections. Too large a setting, however, means wasting resources with idle database connections.

As with thread pools, monitor these settings and adjust them based on your performance tests.

In Tomcat, the connection pools are configured in the Resource elements in `$CATALINA_HOME/conf/Catalina/localhost/ROOT.xml`. Liferay Engineering uses the following configuration during testing:

```
<Resource auth="Container"
    description="Digital Enterprise DB Connection"
    driverClass="com.mysql.jdbc.Driver"
    maxPoolSize="75"   minPoolSize="10"
    acquireIncrement="5"
    name="jdbc/LiferayPool"
    user="XXXXXX"
    password="XXXXXXXXX"
    factory="org.apache.naming.factory.BeanFactory"
    type="com.mchange.v2.c3p0.ComboPooledDataSource"
    jdbcUrl="jdbc:mysql://someServer:3306/liferay_dxp?useUnicode=true
    &amp;characterEncoding=UTF-8&amp;useFastDateParsing=false"/>
```

In this configuration, we start with 10 threads and increment by 5 as needed to a maximum of 75 connections in the pool.

You may choose from a variety of database connection pool providers, including DBCP, C3P0, HikariCP, and Tomcat. You may also choose to configure the Liferay JDBC settings in your portal.properties.

Deactivate Development Settings in the JSP Engine

Many application servers have their JSP Engines configured for development mode by default. Liferay recommends deactivating these settings prior to entering production:

Development mode: This makes the JSP container poll the file system for changes to JSP files. Since you won't be making changes on the fly like this in production, you should turn this off.

Mapped File: Generates static content with one print statement versus one statement per line of JSP text.

To do this in Tomcat, you modify the `$CATALINA_HOME/conf/web.xml` file. Update the JSP servlet to look like the following configuration:

```
<servlet>
    <servlet-name>jsp</servlet-name>
    <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
    <init-param>
        <param-name>development</param-name>
        <param-value>false</param-value>
    </init-param>
    <init-param>
        <param-name>mappedFile</param-name>
        <param-value>false</param-value>
    </init-param>
    <load-on-startup>3</load-on-startup>
</servlet>
```

Thread Pool

Each incoming request to the application server consumes a worker thread for the duration of the request. When no threads are available to process requests, the request is queued to wait for the next available worker thread. In a finely tuned system, the number of threads in the thread pool should be balanced with the total number of concurrent requests. There should not be a significant amount of threads left idle to service requests.

Liferay Engineering recommends an initial setting of 50 threads and then monitoring it within your application server's monitoring consoles. You may wish to use a higher number (e.g., 250) if your average page times are in the 2-3 seconds range. Too few threads in the thread pool may lead to excessive request queuing while too many threads may lead to excessive context switching.

In Tomcat, the thread pools are configured in the Connector element in `$CATALINA_HOME/conf/server.xml`. Further information can be found in the Apache Tomcat documentation. Liferay Engineering used the following configuration during testing:

```
<Connector maxThreads="75" minSpareThreads="50"
    maxConnections="16384" port="8080"
    connectionTimeout="600000" redirectPort="8443"
    URIEncoding="UTF-8"  socketBuffer="-1"
    maxKeepAliveRequests="-1" address="xxx.xxx.xxx.xxx"/>
```

Additional tuning parameters around Connectors are available, including the connector types, the connection timeouts, and TCP queue. Consult the appropriate Tomcat documentation for further details.

39.2 Java Virtual Machine Tuning

Tuning the JVM primarily focuses on tuning the garbage collector and the Java memory heap. These parameters are used to optimize the throughput of your application. We used Oracle's 1.8 JVM for the reference architecture. You may also choose other supported JVM versions and implementations. Please consult the Liferay Digital Enterprise Compatibility Matrix for additional compatible JVMs.

Garbage Collector

Choosing the appropriate garbage collector (GC) helps improve the responsiveness of your Liferay DXP deployment. Liferay recommends using the concurrent low pause collectors:

```
-XX:+UseParNewGC -XX:ParallelGCThreads=16 -XX:+UseConcMarkSweepGC
-XX:+CMSParallelRemarkEnabled -XX:+CMSCompactWhenClearAllSoftRefs
-XX:CMSInitiatingOccupancyFraction=85 -XX:+CMSScavengeBeforeRemark
```

You may choose from other available GC algorithms including parallel throughput collectors and G1 collectors. Liferay recommends first starting your tuning using parallel collectors in the new generation and concurrent mark sweep (CMS) in the old generation.

Note: the value 16 in `ParallelGCThreads=16` varies based on the type of CPUs available. We recommend setting the value according to CPU specification. On Linux machines, you may find the number of available CPUs by running `cat /proc/cpuinfo`.

Note: There are additional "new" algorithms like G1, but Liferay Engineering's tests for G1 have indicated that it does not improve performance. Your application performance may vary and you should add it to your testing and tuning plans.

Code Cache

Java uses a just-in-time (JIT) compiler that generates native code to improve performance. The default size is 48M. This may not be sufficient for larger applications. Too small a code cache reduces performance as the JIT isn't able to optimize high frequency methods. For Digital Enterprise, we recommend starting with 64M for the initial code cache size.

```
-XX:InitialCodeCacheSize=32m -XX:ReservedCodeCacheSize=96m
```

You can examine the efficacy of the parameter changes by adding the following parameters:

```
-XX:+PrintCodeCache -XX:+PrintCodeCacheOnCompilation
```

Java Heap

When most people think about tuning the Java memory heap, they think of setting the maximum and minimum memory of the heap. Unfortunately, most deployments require far more sophisticated heap tuning to obtain optimal performance, including tuning the young generation size, tenuring durations, survivor spaces and many other JVM internals.

For most systems, Liferay recommends starting with at least the following memory settings:

```
-server -XX:NewSize=700m -XX:MaxNewSize=700m -Xms2048m -Xmx2048m -XX:MetaspaceSize=300m
-XX:MaxMetaspaceSize=300m -XX:SurvivorRatio=6 -XX:TargetSurvivorRatio=9 -XX:MaxTenuringThreshold=15
```

On systems that require large heap sizes (e.g., above 4GB), it may be beneficial to use large page sizes. You may activate large page sizes using the following JVM options:

```
-XX:+UseLargePages -XX:LargePageSizeInBytes=256m
```

You may choose to specify different page sizes based on your application profile.

Note: To use large pages in the JVM, you must configure your underlying operation system to activate them. In Linux, run `cat /proc/meminfo` and look at "huge page" items.

Caution: Avoid allocating more than 32GB to your JVM heap. Your heap size should be commensurate with the speed and quantity of available CPU resources.

JVM Advanced Options

The following advanced JVM options were also applied in the Liferay benchmark environment:

```
-XX:+UseLargePages -XX:LargePageSizeInBytes=256m
-XX:+UseCompressedOops -XX:+DisableExplicitGC -XX:-UseBiasedLocking
-XX:+BindGCTaskThreadsToCPUs -XX:UseFastAccessorMethods
```

Please consult your JVM documentation for additional details on advanced JVM options.

Combining the above recommendations together, we have this configuration:

```
-server -XX:NewSize=1024m -XX:MaxNewSize=1024m -Xms4096m
-Xmx4096m -XX:MetaspaceSize=300m -XX:MaxMetaspaceSize=300m
-XX:SurvivorRatio=12 -XX:TargetSurvivorRatio=90
-XX:MaxTenuringThreshold=15 -XX:+UseLargePages
-XX:LargePageSizeInBytes=256m -XX:+UseParNewGC
-XX:ParallelGCThreads=16 -XX:+UseConcMarkSweepGC
-XX:+CMSParallelRemarkEnabled -XX:+CMSCompactWhenClearAllSoftRefs
-XX:CMSInitiatingOccupancyFraction=85 -XX:+CMSScavengeBeforeRemark
-XX:+UseLargePages -XX:LargePageSizeInBytes=256m
-XX:+UseCompressedOops -XX:+DisableExplicitGC -XX:-UseBiasedLocking
-XX:+BindGCTaskThreadsToCPUs -XX:+UseFastAccessorMethods
-XX:InitialCodeCacheSize=32m -XX:ReservedCodeCacheSize=96m
```

Caution: The above JVM settings should formulate a starting point for your performance tuning. Every system's final parameters vary due to many factors including number of current users and transaction speed.

Liferay recommends monitoring the garbage collector statistics to ensure your environment has sufficient allocations for metaspace and also for the survivor spaces. Simply using the guideline numbers above may result in dangerous runtime scenarios like out of memory failures. Improperly tuned survivor spaces also lead to wasted heap space.

39.3 Content Delivery Network

A Content Delivery Network (CDN) is an interconnected system of servers deployed in multiple data centers that use geographical proximity as a criteria to deliver content across the Internet. For more information on CDNs and their general use cases and technical details, see http://en.wikipedia.org/wiki/Content_delivery_network.

Here, you'll first discover the perks of using a CDN in Liferay DXP and learn about general guidelines for using a CDN in your Liferay DXP instance. Then, you'll learn the steps to configure a CDN for your Liferay DXP instance. It's time to distribute your Liferay DXP content around the world!

Using CDN for Performance Enhancements

A CDN serves web resources to users of a Liferay DXP instance. These resources (images, CSS files, JavaScript files, etc.) from the portal are stored on multiple servers around the world. When requested, the resources are retrieved from the server nearest to the user.

The CDN functions as a caching proxy. This means that once static content is copied to a local server, it is stored in a cache for quick and easy retrieval. This drastically improves latency time, because browsers can download static resources from a local server down the street instead of halfway around the world. A user's request to the CDN for content is directed to specific server machine based on an algorithm. The algorithm attempts to find the server closest to the user. The figure below shows a visual representation of using geographical proximity to improve latency.

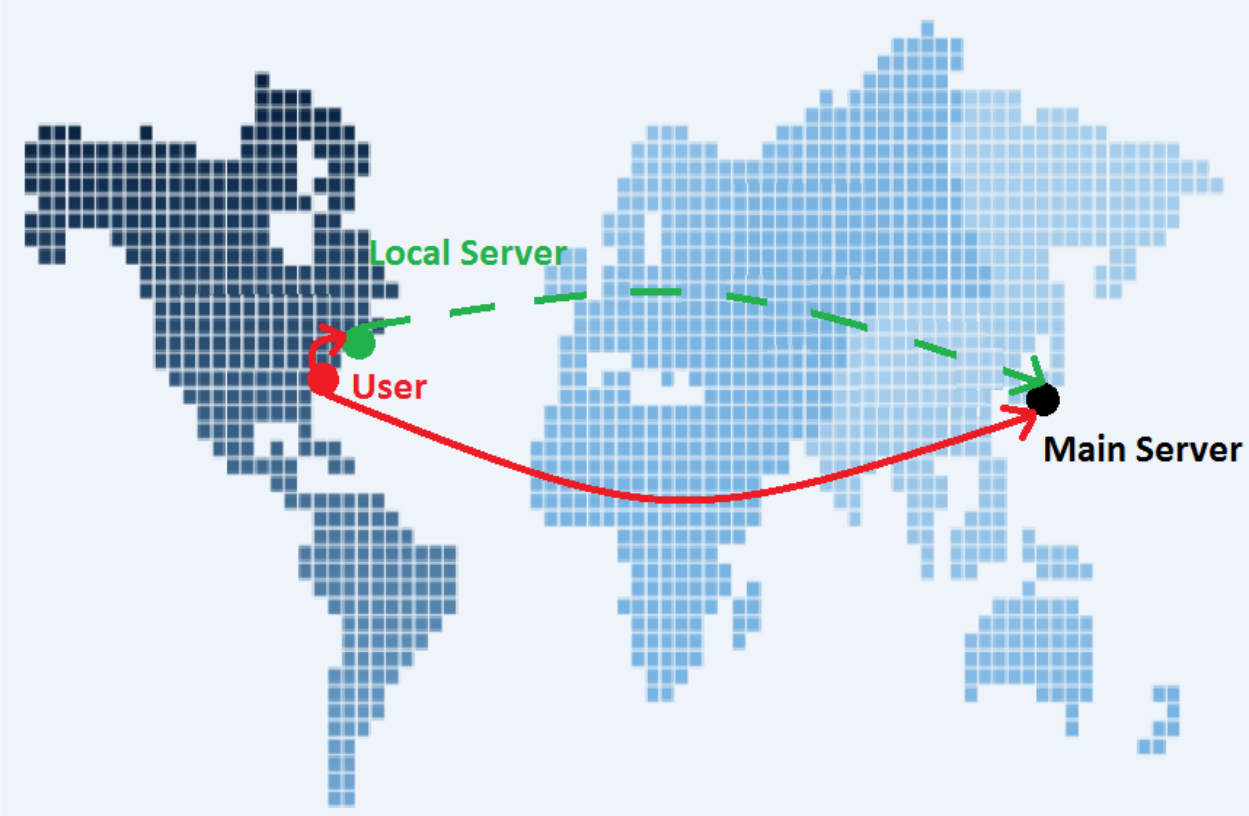


Figure 39.1: The red lines on the map represent the required distances traveled by requests from a server to the user. Using CDN allows a user to request static resources from a much closer local server, improving download times.

Because of the reduced wait time for requests and reduced load on your application server, a CDN is a great option to improve your portal's performance. Using a CDN with Liferay DXP, however, has some restrictions.

Liferay CDN Requirements

Liferay DXP only works with CDNs that can dynamically retrieve requested resources from Liferay DXP. Dynamic resources are resources which change over time or via interaction with end users and thus cannot be cached. For this reason, you should check with your CDN provider to make sure you don't have to manually

upload anything in order for the CDN to work. The CDN must automatically fetch the content from Liferay DXP.

A Liferay DXP-compatible CDN must work like a transparent proxy: A request first goes to the CDN. If the CDN doesn't have the requested resource, the CDN makes an identical request back to the origin (Liferay DXP), caches the resource, then serves the resource.

Once you've configured Liferay DXP to use a CDN (see below), the CDN not only serves portal resources but also plugin resources (e.g., theme resources or JavaScript files referenced from a plugin's `liferay-portlet.xml` file). The CDN only serves resources that are actually included in a plugin. It does not serve resources that are dynamically loaded from external sources.

To get the CDN URL for a resource, developers can simply replace the portal host in the resource path with `themeDisplay.getCDNDynamicResourcesHost()`. Developers should prefix resources with the CDN host name. They should not manually upload any resources to the CDN or put anything on the CDN which requires permission checking or complex policy access.

There are several properties in Liferay DXP that enable you to configure your CDN and tweak it to suite your portal's needs. You'll learn how to do this next.

Configuring Liferay DXP to Use a CDN

Now that you have a general understanding of what a CDN accomplishes and how it's used in Liferay DXP, it's time to set one up for yourself. You can set your CDN and its properties using two different methods:

1. By editing your portal properties file
2. By using the Control Panel

To configure your CDN via properties file, you need to create a `portal-ext.properties` file in your Liferay Home directory and set the appropriate CDN properties. You can view the CDN properties and their descriptions by visiting the Content Delivery Network section of the `portal.properties` HTML document.

Once you configure your CDN host, Liferay DXP generates URLs to the static assets that replace the old host with your new CDN host and so they are automatically cached and served afterwards by the CDN.

To configure your CDN in the Control Panel, navigate to *Control Panel → Configuration → Instance Settings*. In the main configuration, you'll notice three fields related to CDNs:

- *CDN Host HTTP*
- *CDN Host HTTPS*
- *CDN Dynamic Resources Enabled*

These properties are exactly the same as the ones you can specify in your `portal-ext.properties`. Make sure to visit the CDN section of the Properties Document referenced previously if you don't know how to fill in the CDN fields. Make sure to specify your CDN host(s) with a URL that includes the protocol and domain.

Examples,

CDN Host HTTP: `http://cdnhost1.liferay.com`

CDN Host HTTP: `https://cdnhost2.liferay.com`

Once you're finished, click *Save* and your old host is replaced with your new CDN host for static content.

As you can see, configuring a CDN is extremely easy, and can drastically reduce latency time and improve your portal's performance.

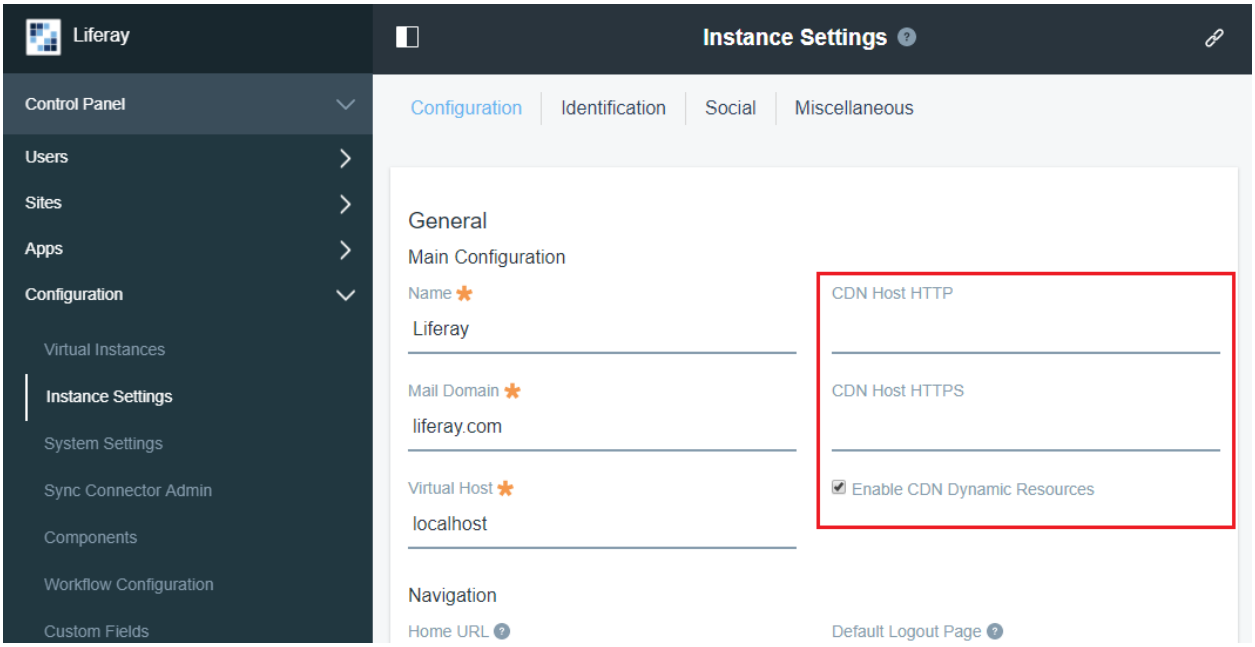


Figure 39.2: The Control Panel lets you configure your portal's CDN.

Chapter 40

Liferay DXP Security

Liferay takes security very seriously. Liferay has established several procedures to make sure that Liferay DXP is as secure as possible. First of all, Liferay DXP is an open source product. As such, Liferay encourages security-minded community members to verify the product they're using. All Liferay users benefit when even a few don't blindly trust the provider! Please read Liferay security statement for more information.

Although we act on community reports, we understands that community reports alone are not enough. Liferay's internal security team also works on improving security. Liferay's internal security team conducts internal security reviews. They check Liferay's source code for common vulnerabilities that can be accidentally introduced by developers. Additionally, all Liferay DXP security related code is reviewed by Liferay's application security team before it's committed. For every major portal release, Liferay works with external security partners to perform security scans and penetration testing.

Because the security cycle never ends, the internal application security team gathers reports from Liferay customers and the Liferay community. The team also monitors other channels (Twitter, the full disclosure mailing list, the liferay.com forums, etc.) to catch every security issue as soon as possible. Once fixed, Liferay's Support, Release, and other teams work on backporting and releasing security patches for all supported versions.

40.1 Liferay DXP Security Overview

Liferay follows the OWASP Top 10 (2013) and CWE/SANS Top 25 lists to ensure that Liferay DXP is as secure as possible. Following these recommendations protects the portal against known kinds of attacks and security vulnerabilities. For example, Liferay DXP's persistence layer is generated and maintained by the Service Builder framework which prevents SQL Injection using Hibernate and parameter based queries.

To prevent Cross Site Scripting (XSS), user-submitted values are escaped on output. To support integration features, Liferay DXP doesn't encode input. Data is stored in the original form as submitted by the user. Liferay DXP includes built-in protection against CSRF attacks, Local File Inclusion, Open Redirects, Uploading and serving files of dangerous types, Content Sniffing, Clickjacking, Path Traversal, and many other common attacks.

To stay on top, Liferay DXP also contains fixes for state-of-the-art attacks and techniques to improve product security. For example, Liferay DXP uses PBKDF2 to store passwords. Liferay DXP also contains mitigation for Quadratic Blowup XXE attack, Rosetta Flash vulnerability, Reflected File Download, and other kinds of attacks.

Authentication Overview

Liferay DXP user authentication can take place using any of a variety of prepared solutions:

- Form authentication using the Sign In Portlet with extensible adapters for checking and storing credentials (portal database, LDAP)
- Single-Sign-On (SSO) solutions - NTLM, CAS, SiteMinder, OpenSSO, OpenID, Facebook
- SAML plugin (https://www.liferay.com/marketplace/-/mp/application/15188711)
- JAAS integration with application server

Note: Although Liferay's SSO solutions are incompatible with WebDAV, they can be used in conjunction with Liferay Sync. See the Publishing Files article for more information on WebDAV and Liferay Sync.

Remote application authentication and authorization can be done using the `AuthVerifier` layer:

- Password based HTTP Basic + Digest authentication
- Token based OAuth plugin
- Portal session based solution for JavaScript applications

Both user authentication and remote application authentication are extensible in Liferay DXP. Developers can create custom Login portlets and plugins, extend the default Login portlet `auth.pipeline`, create `AutoLogin` extensions for SSO, or create custom `AuthVerifier` implementations.

Authorization and Permission Checking

There are several adjustable authorization layers in place to prevent unauthorized or unsecured access to data:

- Remote IP and HTTPS transport check to limit access to Liferay DXP's Java servlets
- Extensible Access Control Policies layer to perform any portal service related authorization check
- Extensible role-based permission framework for almost any portal entity or data (stored in the portal database or elsewhere)
- Portlet Container security checks to control portlet access
- Remote IP check for portal remote API authentication methods
- Service Access Policies to control access to portal remote API

Additional Security Features

Liferay DXP supports other features, too. Liferay DXP users can be assigned to sites, teams, user groups, or organizations. Custom roles can be created, permissions can be assigned to those roles, and those roles can be applied to users. Roles are scoped to apply only with a specific context like a site, an organization, or globally. See Liferay DXP's Roles and Permissions (not yet written) documentation for more information.

Note: Liferay DXP relies on the application server for sanitizing CRLF in HTTP headers. You should, therefore, make sure this is properly configured in your application server, or you may experience false positives in reports from automatic security verification software such as Veracode. There is one exception to this for Resin, which does not have support for this feature. In this case, Liferay DXP sanitizes HTTP headers.

There are additional security plugins available from Liferay Marketplace. For example, you can find an Audit plugin for tracking user actions or an AntiSamy plugin for clearing HTML from XSS vectors.

Liferay DXP provides plenty of configuration options that allow its various security features to be fine-tuned or disabled. Here are a few examples of these kinds of configuration actions:

- Disable the Sign In portlet's *Create Account* link
- Configure Liferay DXP's HTTPS web server address
- Configure the list of allowed servers to which users can be redirected
- Configure the list of portlets that can be accessed from any page
- Configure the file types allowed to be uploaded and downloaded
- Many other options

Secure Development Recommendations

Liferay DXP also provides tools to fight vulnerabilities in code.

For secure development, it's important to have security-mined colleagues in your team. These individuals should consider the security aspects of each stage of the product lifecycle. It's important to start a discussion about security early in the project-planning stage so that threats to user privacy, data, and the system can be identified.

Later, during the implementation phase, developers can use the following list of APIs to address some of the most common vulnerabilities. These APIs should be used consistently across Liferay DXP.

Before releasing a product, it's important to "hack yourself first". I.e., you should conduct penetration tests or source code reviews to catch the low-hanging security-flavored fruit. External penetration testing is also an option for many companies. It's becoming a more popular and less expensive service.

Here's short list of Liferay DXP security APIs:

- `HtmlUtil` - to prevent XSS
- `HtmlUtil#escapeXPath` - prevent XPath injection
- `AuthTokenUtil#checkCSRFToken` - check CSRF tokens
- `FileUtil#createTempFile*` - prevent file system related issues
- `PortalUtil#escapeRedirect` - prevent open redirects
- `StringUtil#random*` - insecure but random enough strings
- `PwdGenerator#getPassword`, `SecureRandomUtil` – cryptographically strong pseudorandom output, optimized for performance
- `PasswordEncryptorUtil` - verification and creation of strong password hashes, configured to use PBKDF2 by default
- `DigesterUtil` - SHA-1 hashes, nowadays usable at most for file checksums

Secure Configuration and Run Recommendations

Liferay DXP is built using the "secure by default" concept in mind. Thus, Liferay DXP's default configuration is already very secure. It's not recommended to disable built-in protections or to allow all values in security white-lists. Such acts may lead to security misconfiguration and an insecure Liferay DXP deployment.

For more information about securing a Liferay DXP installation, please see https://liferay.com/security and https://portal.liferay.dev/people/community-security-team and the resources listed on those pages.

Also, Liferay DXP customers are advised to deploy security patches as described on the customer portal: https://www.liferay.com/group/customer/products/portal/security-vulnerability

For community and CE deployments, the only way to stay secure is to use always the latest community version, which contains all previous security patches. Until a new version is released, the Community Security Team issues patches for the latest CE version via the https://portal.liferay.dev/people/community-security-team page.

Chapter 41

What is SAML?

Liferay DXP's SAML (Security Assertion Markup Language) adapter lets you execute Single Sign On (SSO) and Single Log Off (SLO) in your deployment. Each Liferay DXP instance serves as either the Service Provider (SP) or the Identity Provider (IdP). This article provides the conceptual framework for Liferay DXP's SSO solution.

- Single Sign On
  - Identity Provider initiated SSO
  - Service Provider initiated SSO
- Single Log Off
  - Identity Provider initiated SLO
  - Service Provider initiated SLO

Note: A single Liferay DXP instance is *either* the SP or the IdP in your SSO setup; it can't be both. You can, however, use separate instances for both purposes (for example, one instance is the SP and another is the IdP).

Below is background on how SAML works. To jump right to its configuration, see the articles on Setting Up SAML as an Identity Provider or Setting Up SAML as a Service Provider for instructions on using the SAML adapter. Use the instructions to make the conceptual magic from this article come to life!

Note: If you're migrating from a Liferay SAML adapter prior to version 3.1.0, your properties are automatically migrated to settings. Please see the Configuring SAML article for details on settings.

41.1 Important SAML URLs

For reference, here are a few important SAML URLs.

This URL is the default location of Liferay DXP's metadata XML file:

```
[host]:[port]/c/portal/saml/metadata
```

Note that when configuring SAML for Liferay DXP, no importing of SAML certificates is required. Liferay DXP reads certificates from the SAML metadata XML file. If you want a third-party application like Salesforce to read a Liferay SAML certificate, you can export the Liferay DXP certificate from the keystore. The default keystore file is `[Liferay Home]/data/keystore.jks`. The exported certificate can be imported by a third-party application like Salesforce.

41.2 Single Sign On

Both the IdP and the SP can initiate the Single Sign On process, and the SSO flow is different depending on each one. Regardless of how it's initiated, SSO is configured for HTTPS between the SP and IdP, so all transport-level communication is encrypted. SAML requests are signed using certificates configured in Liferay DXP, using the SAML Web Browser SSO profile as defined in the SAML 2.0 specification.

Consider IdP initiated SSO first.

Identity Provider Initiated SSO

Sometimes a user enters the SSO cycle by sending a request directly from the browser to the IdP.

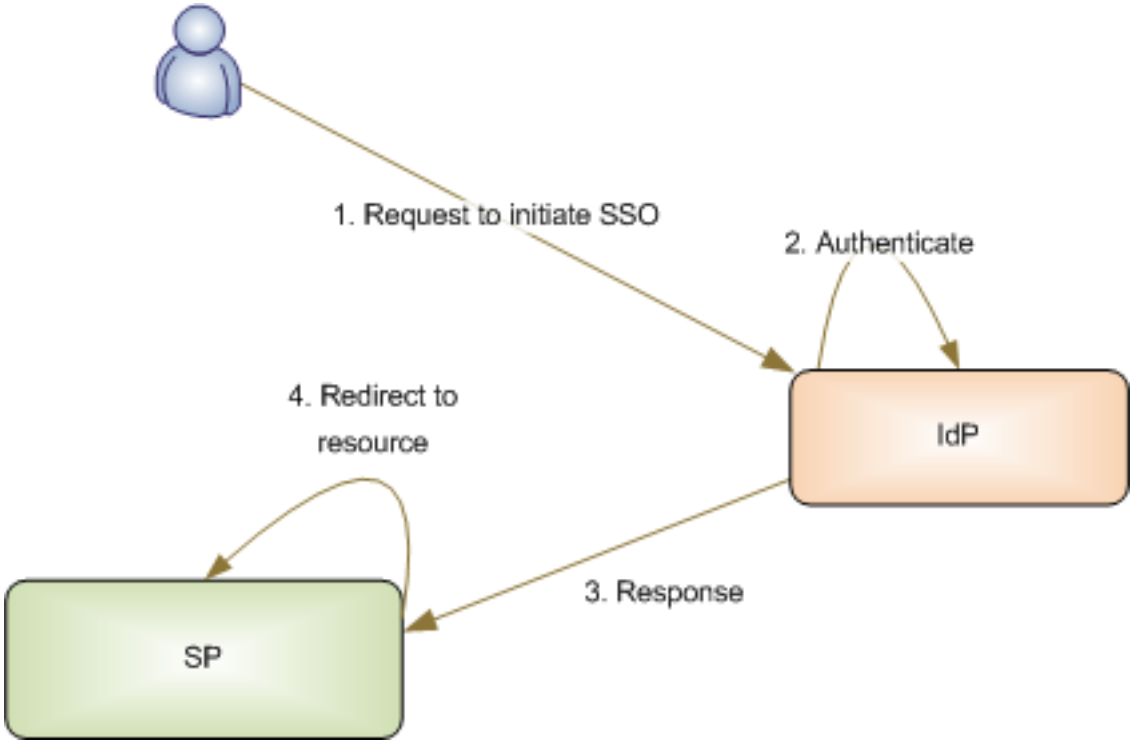


Figure 41.1: Identity Provider Initiated SSO

The SSO Request to the IdP

If Liferay DXP is the IdP, the IdP initiated SSO URL

- Must specify the path as `/c/portal/saml/sso`.
- Must include the `entityId` parameter which is the identifier to a previously configured Service Provider Connection (SPC).

- May include a `RelayState` parameter which contains a URL encoded value to which the user will be redirected upon successful authentication. This URL should point to a location on the desired SPC (according to the SAML 2.0 standards section 3.4.3, this value *must not* exceed 80 bytes in length). It is useful to specify a landing page after SSO has been executed.

For non-Liferay DXP IdPs (Siteminder, ADFS, etc.), consult the vendor's documentation on constructing IdP initiated SSO URLs.

If the IdP determines that the user isn't authenticated, it prompts the user with the appropriate login screen.

The SSO Response from the IdP

Upon successful authentication the IdP constructs a SAML Response. It includes attribute statements configured in the designated Service Provider Connection (SPC; see the next article on setting up the SPC in Liferay DXP's SAML adapter).

The IdP sends the response to the Assertion Consumer Service URL using HTTP-POST or HTTP-Redirect. HTTP-POST is preferred because it reduces the risk that the URL is too long for a browser to handle. Using HTTP-POST, the request contains two parameters:

`SAMLResponse`

and

`RelayState`

Note: The method for sending the SAML response (for example, HTTP-Post) and the Assertion Consumer Service URL are generally imported as part of the SAML metadata XML provided by the SP. In Liferay DXP, you import the SP's metadata in the SAML Adapter's Service Provider Connections tab.

The SP Processes the SSO Response

The SP validates and processes the SAML Response. Liferay DXP's SAML solution requires `SAMLResponse` messages to be signed. This signature process ensures proper identification for the IdP and prevents potential SAML Response spoofing.

- If one Liferay DXP instance is the IdP and another is the SP, make sure the SAML metadata XML file imported into the Liferay DXP SP contains the IdP's certificate.
- If Liferay DXP is the IdP and another application is the SP, export the certificate from the Liferay DXP IdP and import it into the SP's certificate store.

If a `RelayState` is included in the SAML Response, the user is redirected to it. Otherwise the home page of the SP is served.

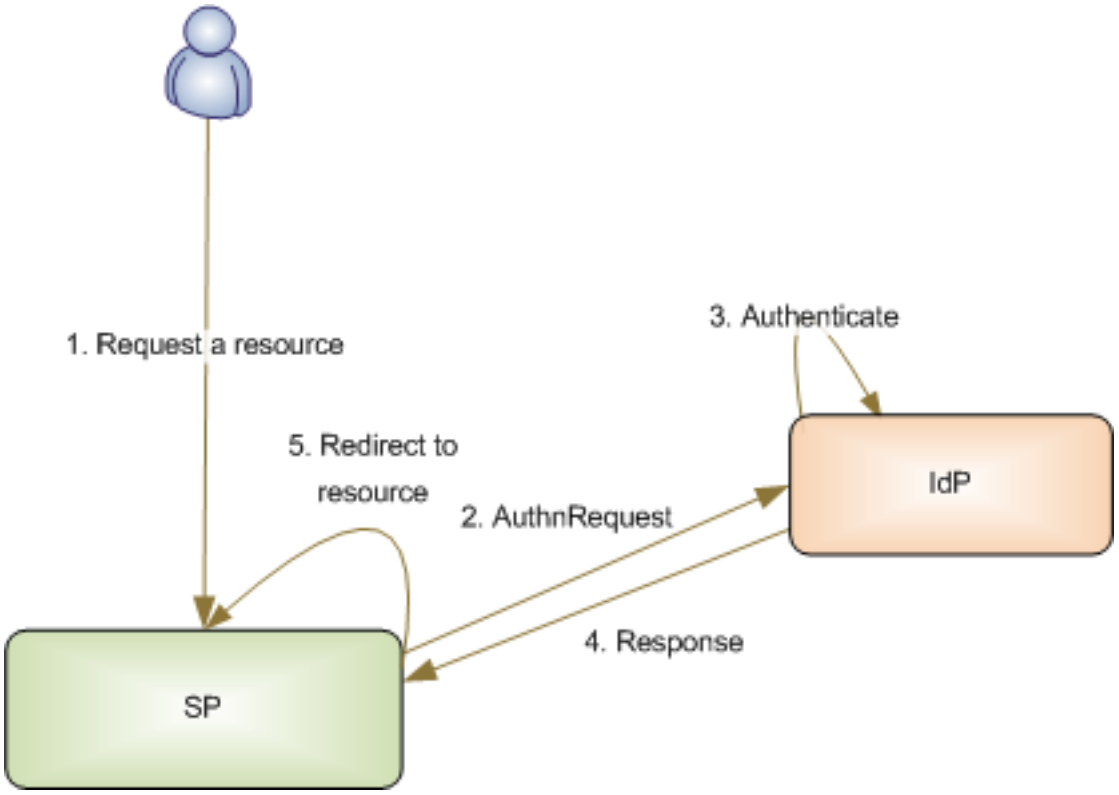


Figure 41.2: Service Provider Initiated SSO

Service Provider Initiated SSO

The SSO Request to the SP

When the user's browser requests a protected resource or sign on URL on the SP, it triggers the SP initiated SSO process. When Liferay DXP is the SAML SP, SSO is initiated either by requesting `/c/portal/login` URL or a protected resource that requires authentication (for example, a document that is not viewable by the Guest role). If the user requests a protected resource, its URL is recorded in the `RelayState` parameter. If the user requested `/c/portal/login`, the `RelayState` can be set by providing the `redirect` parameter. Otherwise, if the portal property `auth.forward.by.last.path` is set to true, the last accessed path is set as the `RelayState`. For non-Liferay DXP SPs, consult the vendor documentation on initiating SSO.

The AuthnRequest to the IdP

The SP looks up the IdP's Single Sign On service URL and sends an `AuthnRequest`. When Liferay DXP is the SP it looks up the configured SAML Identity Provider Connection and sends a SAML `AuthnRequest` to the IdP's Single Sign On service URL as defined in the SAML metadata XML document. Liferay DXP supports sending and receiving the `AuthnRequest` using HTTP-Post or HTTP-Redirect binding. HTTP-Post is preferred.

If the user doesn't have an active session or if `ForceAuthn` was requested by the SP, the user must authenticate by providing his or her credentials. When Liferay DXP is the IdP, authentication occurs in the Login Portlet. Liferay DXP decodes and verifies the `AuthnRequest` before requesting the user to authenticate.

The SSO Response from the IdP

After authentication a SAML Response is constructed, sent to the Assertion Consumer Service URL of the SP, and verified.

When Liferay DXP is configured as the IdP, any attributes configured on the Service Provider Connection are included in the response as attribute statements. The Assertion Consumer Service URL is looked up from the SAML metadata XML of the SP. The response is sent using HTTP-Post or HTTP-redirect binding. The IdP automatically makes this choice based on the SP metadata. HTTP-Post binding is preferred and used when available. HTTP-Redirect binding is fragile because the signature and included assertions often make the URL too long for browsers.

When Liferay DXP is configured as the SP, any response and assertion signatures are verified. Liferay DXP requires the sender to be authenticated. This is done via whole message signature from the issuing IdP. Any responses missing the signature are considered unauthenticated and the response is rejected. The Response can be received via HTTP-Post binding or HTTP-redirect binding. HTTP-Post binding is preferred for the reasons mentioned in the previous section. For non-Liferay DXP SP or IdP vendors, consult their documentation.

The user is redirected to the requested resource or to the URL contained in the `RelayState` parameter (for example, the last page the user accessed before initiating SSO).

41.3 Single Log Off

The Single Log Off request is sent from the user's browser to either the IdP or to a SP, and the SLO flow differs in each case. First consider IdP initiated SLO.

Identity Provider Initiated SLO

The SLO Request to the IdP

An IdP initiated SLO request is a SLO request sent directly to the IdP by the user's browser. When Liferay DXP serves as the IdP, the IdP initiated SSO URL must specify the URL path as

`/c/portal/logout`

If the user is signed on to any configured SP, the SAML plugin takes over the logout process, displaying all the signed on services. The single logout screen displays the authentication status of each SP and whether any SPs can't be logged out of (for example, if the SP is down or doesn't support SLO). For non-Liferay DXP IdPs (Siteminder, ADFS, etc.) consult the vendor's documentation on constructing IdP initiated SLO URLs.

The IdP sends a SAML `LogoutRequest` to the SP.

- When Liferay DXP is configured as the IdP, the `LogoutRequest` is sent using either HTTP-Post, HTTP-Redirect, or SOAP binding. HTTP-Post binding is preferred but in its absence, the first available SLO endpoint with supported binding is selected.
- When Liferay DXP is configured as the SP, supported bindings for `LogoutRequest` are HTTP-Post, HTTP-Redirect, or SOAP.
- For other IdPs or SPs, please consult the vendor's documentation.

The SLO Response from the SP

The SP delivers a `LogoutResponse` to the IdP. When Liferay DXP is configured as the SP, the `LogoutResponse` is delivered using either HTTP-Post, HTTP-Redirect, or direct response to SOAP request. HTTP-Post binding is

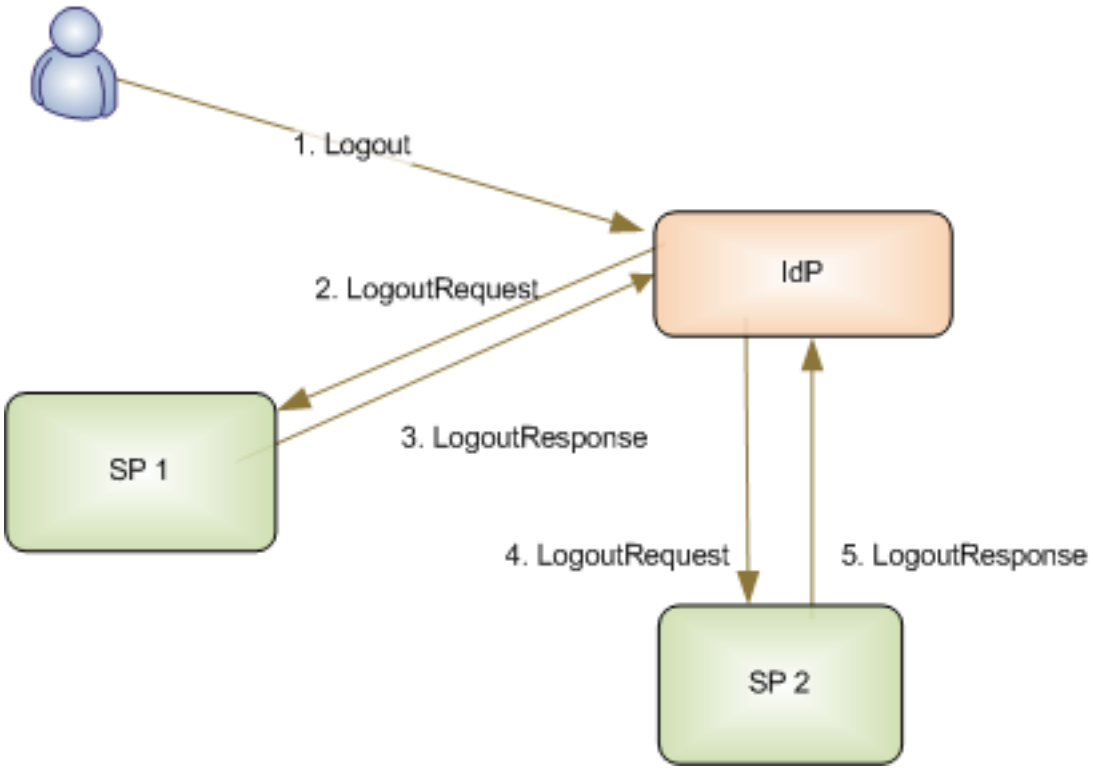


Figure 41.3: Identity Provider Initiated SLO

preferred but in its absence, HTTP-Redirect is used. SOAP is only used to respond to the `LogoutRequest` over SOAP binding.

The IdP sends a SAML `LogoutRequest` to the second SP using either HTTP-Post, HTTP-Redirect, or SOAP binding.

The second SP then delivers the `LogoutResponse` to the IdP using HTTP-Post, HTTP-Redirect, or direct response to SOAP request. The process is repeated for all SPs the user is logged into. When Liferay DXP is the IdP, Liferay DXP logs the user out after the last SP has delivered its `LogoutResponse` or has timed out.

Service Provider Initiated SLO

The SLO Request to the SP

In SP initiated SLO, user's browser requests logout directly to the SP. When Liferay DXP is configured as the SP, the SLO is initiated by requesting the logout URL

```
/c/portal/logout
```

For other SPs, consult the vendor's documentation on initiating SLO.
A SAML `LogoutRequest` is sent to the Single Log Out service URL of the IdP.

- If Liferay DXP serves as the SP, the `LogoutRequest` is sent to the IdP configured by the IdP Connection tab of the SAML provider (see the next article to set up the IdP Connection) and the SLO service URL defined in the SAML metadata. The request is sent using HTTP-POST or HTTP-Redirect binding.
- When Liferay DXP is the IdP, if the user has logged on to other SPs the user is presented with a single logout screen with the status of each SP logout, flagging any that can't be looged out of (some SPs

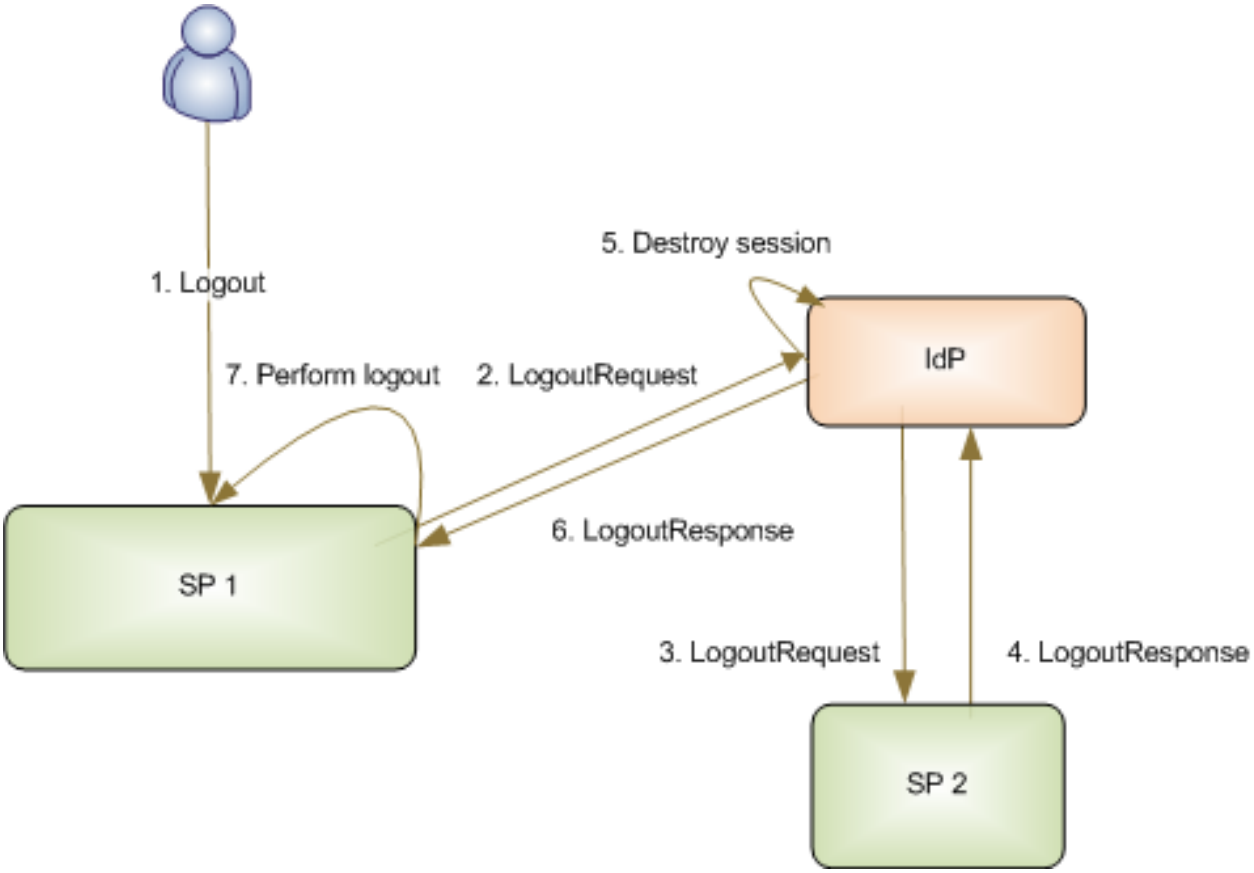


Figure 41.4: Service Provider Initiated SLO

might not support SLO or are currently down). If there are no other SPs to log out of, the SAML session terminates and the IdP destroys its session.

The SLO Response from the SP

If the user is logged in to additional SPs (beyond just the initiating SP), the IdP sends the SAML `LogoutRequest` to each one. When Liferay DXP is the IdP, the `LogoutResponse` is sent using either HTTP-Post, HTTP-Redirect, or SOAP binding.

Each SP delivers its `LogoutResponse` to the IdP. When Liferay DXP is the SP, the `LogoutResponse` is sent using either HTTP-Post, HTTP-Redirect or direct response to SOAP request.

After all additional SPs deliver their `LogoutResponses` to the IdP, the IdP destroys its SSO session. When Liferay DXP is the IdP, once the last SP has delivered its `LogoutResponse` or has timed out, the IdP destroys the Liferay DXP session, logging out the user.

Finally, the IdP sends a `LogoutResponse` to the SP that initiated SLO. The initiating SP terminates its SAML session and logs the user out.

41.4 Related Topics

- Setting Up SAML as an Identity Provider
- Setting Up SAML as a Service Provider

- Token-Based SSO Authentication

41.5 Setting up Liferay DXP as a SAML Identity Provider

An identity provider is a trusted provider that provides single sign-on for users to access other websites. A service provider is a website that hosts applications and grants access only to identified users with proper credentials. SAML is maintained by the OASIS Security Services Technical Committee. Liferay Portal 6.1 EE and later versions support SAML 2.0 integration via the Liferay SAML 2.0 Provider application. It is provided from Liferay Marketplace and allows Liferay DXP to act as a SAML 2.0 identity provider or as a service provider. **Important:** You can set Liferay DXP up as an Identity Provider or as a Service Provider. Each single Liferay DXP instance can serve as an identity provider or as a service provider, but **not both**. Both configurations are covered in this article.

To set Liferay DXP up to act as a SAML Identity Provider, follow the steps below. Before proceeding, note that in step 3 below, you generate a keystore for SAML. This keystore has two storage options:

```
- In the file system
- In the Documents and Media library
```

The file system keystore manager is used by default and the default location is the `[Liferay Home]/data` directory. To use Documents and Media library storage for your keystore instead of file system storage, use the document library keystore manager.

To select a keystore manager, go to *Control Panel → System Settings → SAML KeyStoreManager Implementation Configuration*. There, the options are *Filesystem Keystore Manager* and *Document Library Keystore Manager*.

The great thing about using Document Library storage is that you can use any number of back end file stores. These are protected not only by the system in which you're storing the key, but also by Liferay DXP's permissions system.

Here are the steps for setting up Liferay DXP to act as a SAML Identity Provider:

1. Install the Liferay SAML 2.0 Provider app. To access the SAML Admin interface, click on *Control Panel → Configuration* and then on *SAML Admin*.

2. To begin configuring Liferay DXP to use SAML, select a SAML role for Liferay DXP and choose an entity ID.

   Select the *Identity Provider* SAML role. Enter *liferaysamlidp* if you're setting up an example Liferay DXP instance. Alternatively, choose your own entity ID. Then click *Save*. A new Certificate and Private Key section appears.

3. The Certificate and Private Key section lets you create a keystore for SAML. Click on *Create Certificate* and enter the following information:

   - Your common name (your first and last name)
   - The name of your organization
   - The name of your organizational unit
   - The name of your city or locality
   - The name of your state or province
   - The name of your country
   - The length in days that your keystore will remain valid (how long before the keystore expires)
   - The key algorithm (RSA is the default)
   - The key length in bits (2048 is the default)

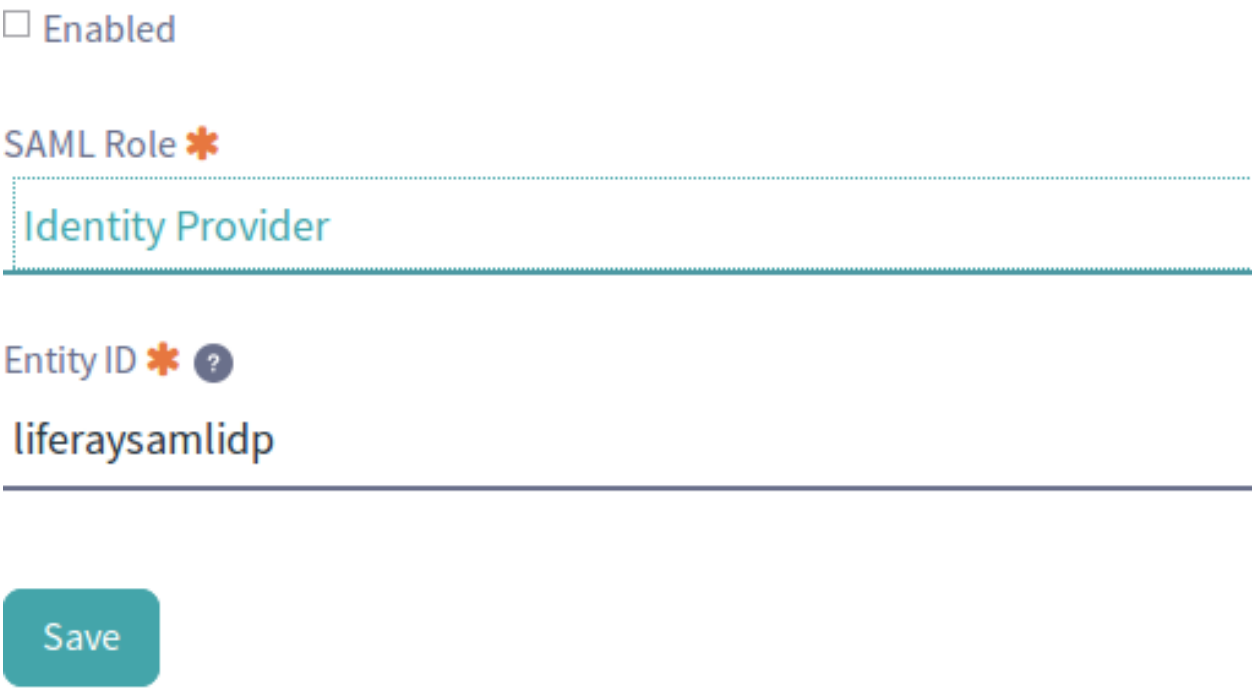


Figure 41.5: Select a SAML role for Liferay and enter an entity ID.

- The key password

When you enter all the required information, click *Save*.

When you create the certificate and private key, you also create a keystore if one doesn't already exist. As described above, this keystore has two storage options: file system storage (the default) and Documents and Media storage.

4. After you click *Save*, you can click *Replace Certificate* at any time to replace the current certificate with a new one if your old one has expired or if you want to change the key's password.

   Three more tabs now appear:

   - *General*: This tab lets you enable or disable a SAML IdP and lets you manage the required keystore.
   - *Identity Provider*: This tab contains other required configurations such as whether to enable SSL. If SSL has been enabled, then SAML requests are not approved unless they are also encrypted.
   - *Service Provider Connections*: This tab manages any Service Providers connected to this Liferay DXP instance.

     See below for more information on the Identity Provider and Service Provider Connections tabs.

5. After you save your certificate and private key information, check the *Enabled* box at the top of the General tab and click *Save*. You successfully set Liferay DXP up as a SAML Identity Provider!

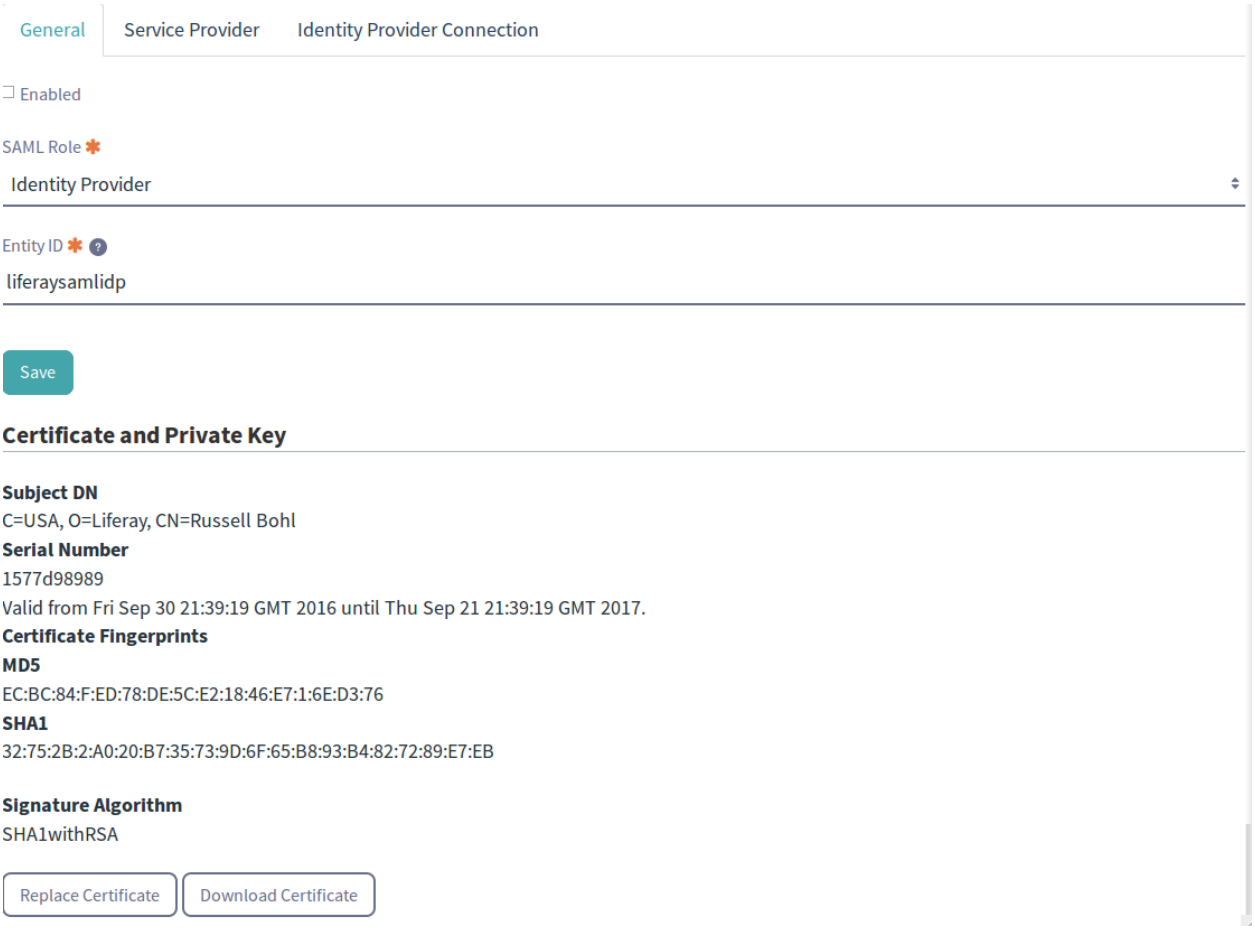


Figure 41.6: The General tab of the SAML Admin portlet displays information about the current certificate and private key and allows administrators to download the certificate or replace the certificate.

Changing the Identity Provider Settings

To configure Liferay DXP's SAML Identity Provider Settings, navigate to the *Identity Provider* tab of the SAML Admin Control Panel entry.

The *Identity Provider* tab includes these options:

Sign Metadata?: When this box is checked, the metadata XML file that's produced is signed.

SSL Required: When this box is checked, any SAML messages that are *not* sent over SSL are rejected. This affects URLs in the generated metadata.

Require Authn Request Signature?: When this box is checked, each Authn Request must be signed by the sending Service Provider. In most cases, this should be enabled.

Session Maximum Age: Specify the maximum duration of the SAML SSO session in seconds. If this property is not set or is set to `0`, the SSO session has an unlimited duration. The SSO session maximum duration can be longer than the portal session maximum duration. If the portal session expires before the SSO session expires, the user is logged back in to Liferay DXP automatically. SSO session expiration does not trigger a single logout from all service providers. You can use the session maximum age, for example, to force users to sign in again after a certain period of time.

Session Timeout: Specify the maximum idle time of the SAML SSO session. Even if the session maximum age is unlimited, the SSO session expires whenever the user's idle time reaches the limit set by the session

timeout property.

Checkpoint

Before adding a Service Provider (SP), verify you've completed these tasks:

1. A SAML keystore has been generated. It can be stored in one of two locations: the data folder or in the Documents and Media library.
2. On the *Identity Provider* tab, the following settings have been set:
    a. **Sign Metadata** has been checked.
    b. **SSL Required** - checked if SSL is active elsewhere. SSL is disabled by default.
    c. **Authn Request Signature Required:** has been checked.
    d. **Session Maximum Age:** has been set. If set to `0`, then the SSO has an unlimited duration.
    e. **Session Timeout:** Specify the maximum idle time of the SAML SSO session.
3. Once the *Enabled* checkbox has been checked, the IdP is now live, and you can generate the required metadata. This URL is the default location of Liferay DXP's metadata XML file:

```
[host]:[port]/c/portal/saml/metadata
```

If this URL does not display correctly, then the SAML instance has not been enabled. Use the URL or click *Save* in the browser to generate an actual `XML` file.

Adding a SAML Service Provider

Of course, setting up Liferay DXP as a SAML Identity Provider is only useful if you can connect to one or more SAML Service Providers. Navigate to the Service Provider Connections tab of the SAML Admin Control Panel entry and click the *Add Service Provider* button to add a SAML Service Provider.

The New Service Provider page includes these options:

Name: The name of the Service Provider with which to connect. The name can be anything; it's purely cosmetic.

Entity ID: The Service Provider's entity ID. This value must match the entity ID declared in the Service Provider metadata.

Enabled: When this box is checked, the Service Provider connection is active.

Assertion Lifetime: Defines the number of seconds after which the SAML assertion issued by the Identity Provider should be considered expired.

Metadata: You can either provide a URL to the Service Provider metadata XML file or you can manually upload the Service Provider metadata XML file. If you provide a URL, the XML file is automatically retrieved and periodically polled for updates. The update interval can be configured in System Settings with the `saml.metadata.refresh.interval` property which specifies a number of seconds. If fetching the metadata XML file by URL fails, you can't enable the Service Provider connection. If the Identity Provider server cannot access the metadata via URL, you can upload the XML file manually. In this case, the metadata XML file is not updated automatically.

Name Identifier Format: Choose the Name Identifier Format used in the SAML Response. This should be set according to what the Service Provider expects to receive. For Liferay Service Providers, any selection other than email address indicates that the Name Identifier refers to screen name. The formats don't have

any special meaning to Liferay Identity Providers. The NameID value is defined by the Name Identifier attribute. See the next option.

Name Identifier Attribute Name: This specifies which attribute of the Liferay DXP User object to use as the NameID value. Possible values include `emailAddress`, `screenName` and `uuid`. Additionally, you can prefix the name with `static:` or `expando:`. If you use the prefix `static`, the value is whatever comes after `static:`. If you use the prefix `expando`, the value is whatever custom field is specified after `expando:`. For example, `expando:SSN` would look up the User custom field with the name `SSN`.

Attributes Enabled: Include and resolve assertion attributes.

Attributes Namespace Enabled: When this box is checked, the attribute names are namespaced like this:

```
urn:liferay:user:expando:
urn:liferay:user:
urn:liferay:groups:
urn:liferay:organizationRole:
urn:liferay:organization:
urn:liferay:roles:
urn:liferay:siteRole:
urn:liferay:userGroupRole:
urn:liferay:userGroups:
```

Attributes: Enter a list of attributes to include in the assertion, one per line. Each line is an expression that gets parsed. Examples:

```
organizations
organizationRoles
roles
siteRoles
userGroups
static:[attributeName]=[attributeValue]
expando:[userCustomFieldName]
```

Note that the full namespace depends on the attribute name. Attribute namespaces can be very useful. Use them when attribute names from different namespaces might conflict. For example, `expando:user` vs `urn:liferay:roles:user`.

Keep Alive URL: If users are logged into several Liferay DXP SP instances via a Liferay DXP IdP, their sessions can be kept alive as long as they keep a browser window open to one of them. Configure this only if the SP is Liferay DXP. The URL is `https://[SP host name]/c/portal/saml/keep_alive`.

Checkpoint

Verify your settings are correct when connecting the Liferay DXP-based IdP to its first SP. SPs connect to only one IdP, so if the first one doesn't work, the rest won't either.

1. Provide a general name for the SP.
2. The `Entity ID` name must be identical to the one declared in the Service Provider metadata.
3. Check the *Enabled* checkbox.
4. Set a value for the *Assertion Lifetime*.
5. Make sure the SP's metadata has been provided either as a URL or an XML file has been uploaded.
6. Make sure *Name Identifier Format* and *Name Identifier Attribute Name* have been set.
7. Make sure *Attributes Namespace Enabled* has been set.

If you don't have a Service Provider to add right now, that's fine. In the next section, you'll learn how to set Liferay DXP up as a SAML Service Provider. After you set up another Liferay DXP instance as a Service Provider, come back to this Liferay DXP installation and add the Service Provider: *Control Panel → SAML Admin → Service Provider Connections → Add Service Provider*.

41.6 Setting up Liferay DXP as a SAML Service Provider

Many of these steps are similar to configuring Liferay DXP as a SAML Identity Provider. As a reminder, a single Liferay DXP installation can be configured as a SAML Identify Provider *or* as a SAML Service Provider but not as both. If you already set up one Liferay DXP installation as a SAML Identity Provider, use a *different* Liferay DXP installation as a SAML Service Provider.

Note: If you're using a third party IdP with Liferay DXP as the SP, all messages coming from the IdP must be signed. If they're not, an error message appears and communication between the IdP and Liferay DXP fails.

---

1. Install the Liferay SAML 2.0 Provider app. To confirm that the app was successfully deployed, look for the *SAML Admin* entry in the Configuration section of the Control Panel.

2. To begin configuring Liferay DXP to use SAML, you must select a SAML role for Liferay DXP and you need to choose an entity ID. Select the *Service Provider* SAML role. Enter *liferaysamlsp* if you're setting up an example Liferay DXP installation. Alternatively, choose your own entity ID. Then click *Save* and a new section entitled Certificate and Private Key appears.

3. The Certificate and Private Key section is for creating a keystore for SAML. Click *Create Certificate* and enter the following information:

    - Your common name (your first and last name)
    - The name of your organization
    - The name of your organizational unit
    - The name of your city or locality
    - The name of your state or province
    - The name of your country
    - The length in days that your keystore will remain valid (how long before the keystore expires)
    - The key algorithm (RSA is the default)
    - The key length in bits (2048 is the default)
    - The key password

    When you enter all the required information, click *Save*.

4. After you clicked *Save*, check that you can view information about your certificate or download your certificate. If you can, you successfully created a keystore. After you create a keystore, additional options appear. There are three tabs:

    - *General*: This tab enables or disables SAML IdP and manages the required keystore.
    - *Service Provider*: This tab manages basic and advanced configurations for the SP.
    - *Identity Provider Connection*: This tab manages connections to the IdP. There can be only one IdP connection.

Note that these options are different than if you were setting up Liferay DXP as an Identity Provider.

5. Next, you need to configure an Identity Provider connection. Click on the *Identity Provider Connection* tab. Enter a name for the Identity Provider, enter its entity ID, and enter its metadata URL. If you have already followed the previous instructions and configured a separate Liferay DXP installation as an Identify provider, you'd enter the following information:

    - Name: *Liferay IdP*
    - Entity ID: *liferaysamlidp*
    - Clock Skew
    - Force Authn
    - Metadata URL: http://localhost:8080/c/portal/saml/metadata (test this URL first)
    - Name Identifier Format
    - Attribute Mapping
    - Keep Alive URL

    Important: The Liferay SAML 2.0 Provider app supports using *either* a URL to a SAML IdP metadata file *or* an actual (uploaded) SAML metadata XML file. The value entered in the *Metadata URL* field will only be persisted to the database when there is one entered metadata URL and there is no specified metadata XML file. Otherwise, Liferay DXP keeps the original metadata URL in the database. This behavior ensures that once a metadata URL has been specified, there will always be a metadata URL saved in the database. This way, if a portal administrator forgets the previously entered metadata URL or its format, he or she can simply look at the displayed metadata URL and either choose to modify the displayed metadata URL or to overwrite the previously saved metadata URL by specifying a metadata XML file.

    Currently, the SAML Provider app does not provide a way to "clear" the SAML IdP metadata URL or metadata XML file fields using the Control Panel UI. If you really need to clear these fields, it's possible (but not recommended) to delete the contents of the SAML IdP metadata URL and metadata XML file columns of the `SamlSpIdpConnection` table of Liferay DXP's database.

6. Finally, after you save your certificate and private key information and configure an Identity Provider connection, check the *Enabled* box at the top of the General tab and click *Save*. Liferay DXP is now a SAML Service Provider!

Note that the SAML Service Provider session is tied to the normal session on the application server. Session expiration on the application server terminates the session on the Service Provider but does not initiate single logout.

Checkpoint

1. A SAML keystore has been generated.
2. Verify the connection to the IdP.
    a. *Name*: generic name for the IdP.
    b. *Entity ID*: the same name of the IdP. If the IdP is another Liferay DXP instance, then it is the same name as the above example.
    c. *Metadata URL*: The IdP's metadata as a URL or as an XML file.

d. If the IdP is another Liferay DXP instance, ensure its corresponding Service Provider Connection for this SP is enabled.

3. On the *General* tab, the *Enabled* checkbox has been checked.

4. Once *Enabled* checkbox has been checked, the service provider's metadata becomes available:

```
[host]:[port]/c/portal/saml/metadata
```

Changing the SAML Service Provider Settings

If you'd like to configure Liferay DXP's SAML Service Provider Settings, navigate to the Service Provider tab of the SAML Admin portlet.

The Service Provider tab includes these options:

Require Assertion Signature?: When this box is checked, SAML assertions must be individually signed in addition to the entire SAML message.

Note: Because Liferay requires the SAML response to be signed, individual assertions need not be signed. The SP and IdP should always communicate over `https` to have encryption at the transport level.

If you believe man-in-the-middle attacks are possible, the SAML response can be signed.

Clock Skew: Clock skew is a tolerance in milliseconds used by the Service Provider for verifying expiration of messages and assertions. This can be used to mitigate time differences between the clocks of the Identity Provider and the Service Provider. This usually only matters when assertions have been made to expire very quickly.

LDAP Import Enabled: When this box is checked, user information is imported from the configured LDAP connection based on the resolved NameID. LDAP connections can be configured from Instance Settings.

Sign Authn Requests: When this box is checked, the AuthnRequest is signed even if the Identity Provider metadata indicates that it's not required.

Sign Metadata: When this box is checked, the metadata XML file is signed.

SSL Required: When this box is checked, any SAML messages that are not sent over HTTPS are rejected. This does not affect how URLs are generated.

Changing the SAML Identity Provider Connection Settings

If you'd like to configure Liferay DXP's SAML Identity Provider Settings, navigate to the Identity Provider Connection tab of the SAML Admin portlet.

Name: The name of the Identity Provider with which to connect.

Entity ID: The Identity Provider's entity ID. This value must match the entity ID declared in the Identity Provider metadata.

Clock Skew: Clock skew is a tolerance in milliseconds used by the Service Provider for verifying expiration of messages and assertions. This can be used to mitigate time differences between the clocks of the Identity Provider and the Service Provider. This usually only matters when assertions have been made to expire very quickly.

Force Authn: When this box is checked, the Service Provider asks the Identity Provider to re-authenticate the user before verifying the user.

Metadata: You can either provide a URL to the Identity Provider metadata XML file or you can manually upload it. If you provide a URL, the XML file is automatically retrieved and periodically polled for updates.

You can change the update interval in System Settings by modifying the `saml.metadata.refresh.interval` property which specifies a number of seconds. If fetching the metadata XML file by URL fails, you can't enable the Identity Provider connection. If the metadata is inaccessible via URL, you can upload the XML file manually. In this case, the metadata XML file is not updated automatically.

Name Identifier Format: Choose the Name Identifier Format used in the SAML Response. This should be set according to what the Service Provider expects to receive. For Liferay Service Providers, any selection other than email address indicates that the Name Identifier refers to screen name. The formats don't have any special meaning to Liferay Identity Providers. The NameID value is defined by the Name Identifier attribute.

Attribute Mapping: The attribute mapping is done from the attribute name or friendly name in the SAML Response to the Liferay DXP attribute name. For example, if you want to map a response attribute named `mail` to the Liferay DXP attribute `emailAddress`, you'd enter the following mapping:

```
mail=emailAddress
```

Available Liferay DXP attributes are: `emailAddress`, `screenName`, `firstName`, `lastName`, `modifiedDate`, and `uuid`.

Keep Alive URL: If users are logged into several Liferay DXP SP instances via a Liferay DXP IdP, their sessions can be kept alive as long as they keep a browser window open to one of them. Configure this only if the IdP is Liferay DXP. The URL is `https://[IdP host name]/c/portal/saml/keep_alive`. On the Liferay DXP IdP, configure this URL the same way, but point back to this SP.

Save your changes when you are finished configuring the Liferay DXP instance as a service provider. There is no need to restart the server and the changes will be applied immediately.

The previous two sections explained how to use the SAML 2.0 Provider app's Control Panel interface to configure Liferay DXP as an Identity Provider or as a Service Provider. Such configurations should only be made through the SAML Control Panel interface and not via properties. Some features of the Liferay SAML 2.0 Provider app are not available as properties.

Limitation: The Liferay SAML app can only be used with a single virtual host. Technically, this means that in the SAML metadata for Liferay DXP, only one binding can be added in this form:

```
<md:EntityDescriptor>
...
<md:SPSSODescriptor>
...
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://portal.domain.com/c/portal/saml/acs" index="1" isDe
...
</md:SPSSODescriptor>
</md:EntityDescriptor>
```

Setting Up Liferay DXP as a SAML Service Provider in a Clustered Environment

If you want to use the Liferay SAML 2.0 Provider app as an SSO solution for a clustered Liferay DXP environment, follow the steps in this section. Before proceeding, make sure that the following assumptions apply to your scenario.

If you're running a multi-node cluster behind a load balancer, follow these steps to enable all the nodes as SPs.

Before you begin, consider the type of keystore manager you want your cluster to use.

To select a keystore manager, go to *Control Panel → System Settings → SAML KeyStoreManager Implementation Configuration*. There, the options are *Filesystem Keystore Manager* and *Document Library Keystore Manager*.

All nodes in the cluster should be configured to use the same Keystore Manager.
If using the Filesystem Keystore Manager (the default):

1. Configure each node of your Liferay DXP cluster as a SAML service provider using the instructions of the previous section.
2. Copy the keystore file (`[Liferay Home]/data/keystore.jks`, by default) from the first Liferay DXP node to the remaining Liferay DXP nodes. This file is the Java keystore that's created by the SAML Provider app. The keystore contains the valid or self-signed certificate managed by the SAML Provider app.
3. Verify that the service provider metadata has been generated to be used either as a URL or an XML file. The metadata is the same for all nodes because of the same database back-end. The IdP's request goes through the load balancer.
4. At this point, all the Liferay DXP nodes have the same SAML SP configuration and each of them can respond to web requests and handle the SAML protocol. To test your SSO solution, sign into Liferay DXP via your load balancer, navigate to a few pages of a few different sites, and then log out.

If using the Document Library Keystore Manager, skip step 3 because the keystore file is stored in the database shared by all the nodes.

Now you know how to configure Liferay DXP either as a SAML identity provider or a service provider. You also know how to configure SAML in a clustered environment.

41.7 Configuring SAML

As noted in the previous tutorials, anything related to configuring SP connections must be done through the SAML Admin UI where configurations are saved to Liferay's database. SP connections can no longer be made via properties files as they were in versions prior to 3.1.0.

This is an portal instance scoped configuration which can be managed via OSGi Configuration Admin. The affected properties are those in the `SAMLProviderConfiguration` metatype:

```
- `saml.keystore.credential.password`
- `saml.sp.assertion.signature.required`
- `saml.idp.authn.request.signature.required`
- `saml.sp.clock.skew`
- `saml.default.assertion.lifetime`
- `saml.sp.default.idp.entity.id`
- `saml.enabled`
- `saml.entity.id`
- `saml.sp.ldap.import.enabled`
- `saml.role`
- `saml.idp.session.maximum.age`
- `saml.idp.session.timeout`
- `saml.sp.sign.authn.request`
- `saml.sign.metadata`
- `saml.ssl.required`
- `saml.idp.metadata.name.id.attribute`
```

The SAML Admin UI remains the place for creating the portal instance scoped configuration instances.

Note: Don't use OSGi `.config` files or Liferay DXP's System Settings Control Panel application to configure SAML providers (IdP or SP). The System Settings UI is auto-generated, and is for advanced admins. It does not perform the enhanced validation on the fields that the SAML Admin UI performs, so it could allow administrators to create invalid configurations.

Note that there is also a system wide configuration, represented by the `SamlConfiguration` metatype.

If you used Liferay 6.2, please note that the following system wide properties were removed:

```
`saml.metadata.paths` (served no purpose after removal of SP connection defaults)
`saml.runtime.metadata.max.refresh.delay`
`saml.runtime.metadata.min.refresh.delay`
```

The latter two properties were replaced with the single property `saml.runtime.metadata.refresh.interval`.

Note also the introduction of the *SAML KeyStoreManager Implementation Configuration* in *Control Panel → System Settings*. The options for this configuration are explained above in the Setting up Liferay DXP as a SAML Identity Provider section.

41.8 Logging in to Liferay DXP

One of the primary functions of a web portal is to restrict access to different pages, content, and web applications. These kinds of portal resources should only be accessible by the appropriate users. E.g., a student who logs in to a university portal should not be able to access the same resources that are available to a professor. Similarly, a patient who logs in to a health care portal should not be able to access the same resources that are available to a doctor. Some portal resources (at least a login page) should be available to users who have not logged in. In Liferay DXP, users who have not logged in are called *guest* users.

Liferay DXP's Sign In portlet provides the basic means for users to log in to Liferay DXP. By default, users can also use the Sign In portlet to create new accounts or to request a password reset. The home page of a default Liferay DXP installation contains a Sign In portlet. You can access this page at http://localhost:8080/web/guest/home if you're running Liferay DXP locally.

Even if the Sign In portlet has not been added to any Liferay DXP page that's accessible to a guest user, users can still access it by navigating to this URL:

- localhost:8080/c/portal/login

Note that Liferay DXP's configured authentication type determines the type of credentials that the user needs to enter in order to log in. Liferay DXP supports three authentication types: authentication by email address, screen name, or user ID. To choose an authentication type, navigate to the Control Panel, click on *Configuration → Instance Settings → Authentication* and use the *How do users authenticate?* dropdown to make a selection. Alternatively, add the following lines to your `portal-ext.properties` file, uncomment the appropriate line, comment out the others, and restart your Liferay DXP server.

```
company.security.auth.type=emailAddress
#company.security.auth.type=screenName
#company.security.auth.type=userId
```

Liferay DXP's default authentication type uses users' email addresses. Users' screen names or user IDs can be used instead. Screen names are chosen when a new account is created either by the user or by an administrator. User IDs are autogenerated by Liferay DXP. Regardless of which authentication type is configured, users must always enter a password to log in to Liferay DXP.

By default, guest users can create accounts on your portal by clicking on the *Create Account* link in the Sign In portlet, completing the form, and submitting it. If a user already has an account but has forgotten its password, the user can click on the *Forgot Password* link to request a password reset. Both the *Create Account* form and the *Forgot Password* form include a CAPTCHA-based text verification field. Using CAPTCHA prevents bots from submitting these forms. Liferay DXP can be configured to use reCAPTCHA instead of

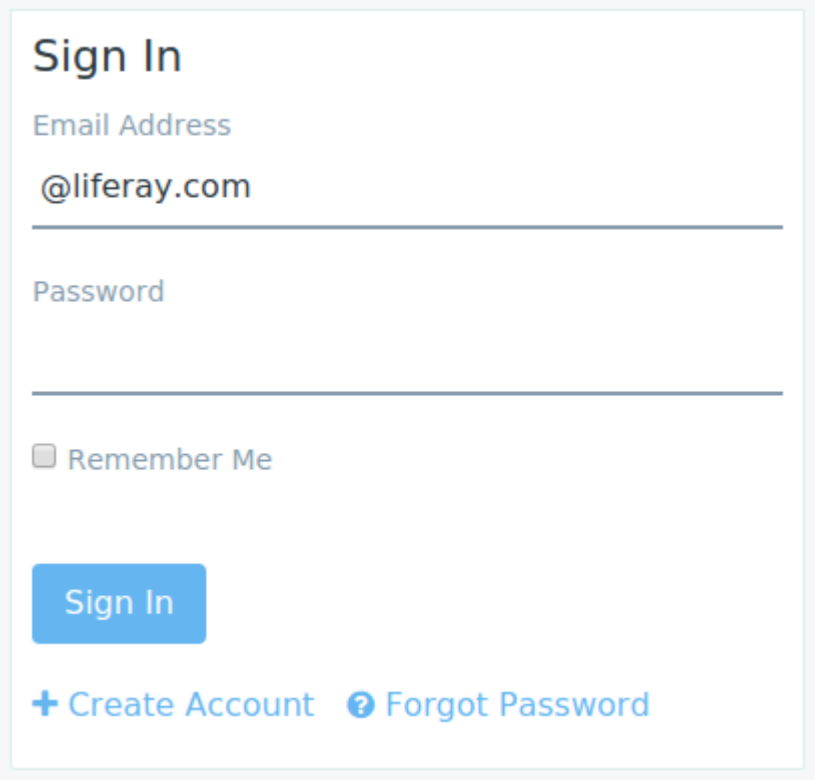


Figure 41.7: By default, the Sign In portlet allows users to log in, create a new account, or request a password reset.

CAPTCHA. One advantage of using reCAPTCHA is that it can allow visually impaired users to pass the test. To configure Liferay DXP to use reCAPTCHA, navigate to the Control Panel, then click on *System → Server Administration → CAPTCHA*.

The security requirements of some web portals require that all user accounts be created by administrators. Liferay DXP supports this use case by allowing the *Create Account* to be removed. To prevent guest users from creating new user accounts, navigate to the Control Panel, click on *Configuration → Instance Settings → Authentication* and uncheck the *Allow strangers to create accounts?* box. You can also disallow users from requesting forgotten passwords or from requesting password reset links by unchecking the appropriate boxes.

Remember that using the Sign In portlet provides the most basic way for users to log in to Liferay DXP but it's not the only way. Liferay DXP allows user accounts to be imported from and exported to LDAP directories. Liferay DXP can be configured to use single-sign-on (SSO) solutions. Liferay DXP supports token-based authentication. This mechanism allows remote web applications to authenticate to Liferay DXP. Please refer to the other articles in this section for more information. Finally, remember that Liferay DXP's user authentication and remote application authentication mechanisms are extensible.

41.9 Service Access Policies

Service access policies are a new feature in 7.0. They are an additional layer of web service security defining services or service methods that can be invoked remotely. Many of them can be applied at once to produce a combined effect. Service access policies apply only to remote services, not to local services. To help you understand how service access policies fit into the big picture, here's a summary of Liferay DXP's web service

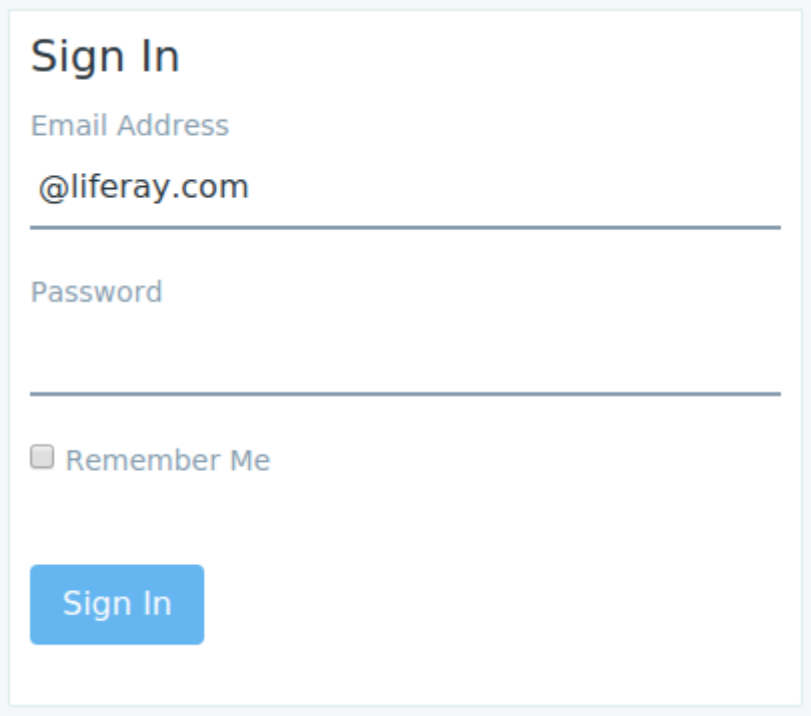


Figure 41.8: Here's a view of the Sign In portlet with the *Create Account* and *Forgot Password* options removed.

security layers:

IP permission layer: The IP address from which a web service invocation request originates must be white-listed in the Liferay DXP server's portal properties file. Any attempted web service invocation coming from a non-whitelisted IP address automatically fails.

Service access policy layer: The method corresponding to a web service invocation request must be whitelisted by each service access policy that's in effect. Wildcards can be used to reduce the number of service classes and methods that must be explicitly whitelisted.

Authentication/verification layer (browser-only): If a web service invocation request comes from a browser, the request must include an authentication token. This authentication token is the value of the `p_auth` URL parameter. The value of the authentication token is generated by Liferay DXP and is associated with your browser session. The `p_auth` parameter is automatically supplied when you invoke a Liferay DXP web service via the JSON web services API page or via JavaScript using `Liferay.Service(...)`. If Liferay DXP cannot associate the caller's authentication token with a portal user, the web service invocation request fails.

User permission layer: Properly implemented web services have permission checks. The user invoking a web service must have the appropriate Liferay DXP permissions to invoke the service.

Note that service access policies respect Liferay DXP's permissions system. Even if a service access policy grants a user access to a remote service, the user must still have the appropriate permissions to invoke that service.

Service access policies are especially useful when remote applications such as mobile devices or Liferay Sync instances need to access Liferay DXP's web services. Your portal administrators can use service access policies to ensure that these devices can only invoke remote services from approved lists that can be modified at runtime.

Managing Service Access Policies

To manage service access policies, navigate to Liferay DXP's Control Panel and click on *Service Access Policy* under the Configuration heading. Here, you can see the default service access policies and you can add new ones. When creating or editing service access policies, keep these points in mind:

- Service access policy names must be unique per portal instance.
- Service access policy names can include only these allowed characters:

  ```
  0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz#:@-./_
  ```

- Service access policy titles can be localized; service access policy names cannot be localized.
- Allowed service signatures must be entered one per line. Wildcards (*) are allowed for both class names and method names. The # symbol must be used to separate a class name from a method name. For example,

  ```
  com.liferay.portal.kernel.service.UserService
  ```

  allows any method from the `UserService` class to be invoked.

  ```
  com.liferay.document.library.kernel.service.DLAppService#get*
  ```

  allows any method from the `DLAppService` that starts with `get` to be invoked. Thus,

  ```
  com.liferay.portal.kernel.service.UserService
  com.liferay.document.library.kernel.service.DLAppService#get*
  ```

  allows any method from the `UserService` class to be invoked and any method from the `DLAppService` whose name starts with `get` to be invoked.

Liferay DXP contains four service access policies that are enabled by default:

- SYNC_DEFAULT
- SYNC_TOKEN
- SYSTEM_DEFAULT
- SYSTEM_USER_PASSWORD

The `SYSTEM_DEFAULT` policy applies to every request, including unauthenticated requests. Thus, its list of allowed method signatures is empty, meaning that no Liferay DXP API can be invoked. The `SYSTEM_USER_PASSWORD` policy applies to requests for which `AuthVerifierResult.isPasswordBasedAuthentication` is true: i.e., whenever user authentication took place using a password. Its list of allowed method signatures is *, meaning that any Liferay DXP API can be invoked. Of course, since Liferay DXP API functions include permission checks, the success of an invocation depends on whether the user has the required permission. If you want to completely disallow certain Liferay DXP API functions from being invoked, you can change the `SYSTEM_USER_PASSWORD` policy to something more restrictive than *.

The `SYNC_DEFAULT` policy applies to every Liferay Sync request, including unauthenticated Sync requests. Its list of allowed method signatures includes only the `com.liferay.sync.service.SyncDLObjectService.getSyncContext` method. The `SYNC_TOKEN` policy applies to Sync requests which are accompanied by an authentication token.

Its list of allowed signatures includes `com.liferay.sync.service.*`, meaning that any Liferay DXP API function that's a method of a class in this package can be invoked.

`SYNC_DEFAULT` and `SYSTEM_DEFAULT`, as their names suggest, are default service access policies. Default service access policies are applied to all incoming requests, including unauthenticated requests. Administrators can create new default service access policies. To add, edit, or delete a service access policy, navigate to the *Configuration → Service Access Policy* section of the Control Panel. Each Liferay plugin can declare own default policy. (The `SYNC_DEFAULT` policy is a good example.) This policy can then be changed or disabled by portal administrator. In this case, the plugin can still verify that the policy exists so there is no need to redefine or update it.

By default, Liferay's tunneling servlet uses the `SYSTEM_USER_PASSWORD` service access policy. You can, however, create your own policy for the tunneling servlet and use the property `service.access.policy.name` for the `TunnelingServletAuthVerifier` to specify that your policy should be used instead.

Service Access Policy Module

Liferay's service access policy functionality is provided by the Service Access Policy module. This module includes the following important classes:

- `com.liferay.portal.kernel.security.service.access.policy.ServiceAccessPolicy`: defines the public interface for `ServiceAccessPolicy`.
- `com.liferay.portal.kernel.security.service.access.policy.ServiceAccessPolicyManager`: defines the public interface for retrieving instances of `ServiceAccessPolicy`.
- `com.liferay.portal.kernel.security.service.access.policy.ServiceAccessPolicyManagerUtil`: bridges service access policy functionality to the parts of Liferay's core that have not yet been modularized.
- `com.liferay.portal.kernel.security.service.access.policy.ServiceAccessPolicyThreadLocal`: makes `ServiceAccessPolicy` instances active.

Liferay's Service Access Policy module resides in the `modules/apps/service-access-policy` directory in Liferay DXP's source code. In a running Liferay DXP instance, the service access policy functionality is provided by these three bundles which you can find in the `[Liferay Home]/osgi/modules` directory:

- `com.liferay.service.access.policy.api.jar`
- `com.liferay.service.access.policy.service.jar`
- `com.liferay.service.access.policy.web.jar`

These modules provide the service access policy management UI that's accessible from the Control Panel. They also provide the interface and default implementation for `ServiceAccessPolicy`.

To configure the Service Access Policy module, navigate to Liferay DXP's Control Panel, click on *System Settings*, and find the *Service Access Policies* module in the Foundation section. Click on its name to edit it. Here, you can edit the default service access policy configuration. You can also force a default policy to be applied even when no policies are applied by the `AuthVerifier`.

Liferay DXP also contains an `AuthenticatedAccessControlPolicy`. This policy doesn't do anything if a `ServiceAccessPolicyManager` implementation is present. If the service access policy module is disabled, however, the `AuthenticatedAccessControlPolicy` provides a fallback that still requires authenticated access for web services.

Summary

Great! Now you know service access policies can restrict access to Liferay DXP's web services. Custom service access policies can be created by portal administrators. They are applied by the portal's token authenticator, e.g., by OAuth. The service access policies attached to an application define the services that can be invoked by the application. For example, a Sync service access policy could be created that defines all of the services that the Sync application needs to invoke. Then this service access policy could be attached to the Sync application on OAuth.

Related Topics

(Coming Soon)

41.10 Authentication Verifiers

Liferay DXP includes a centralized and extensible authentication layer called the authentication verification layer. This layer is mainly used for authenticating remote invocations of Liferay DXP's API.

The main responsibilities of the authentication verification layer are to

1. Verify provided credentials using registered `AuthVerifier` instances
2. Create portal authorization contexts based on verification results

If no available `AuthVerifier` is able to verify request credentials, an authorization context supporting non-authenticated access is created for a guest user. This allows each Liferay DXP API to expose only a single API endpoint. In contrast, legacy (prior to 6.2) versions of Liferay DXP exposed two API endpoints for each API: the `/api/endpoint` URI was for non-authenticated access and the URI `/api/secure/endpoint` was for authenticated access.

Liferay DXP offers built-in `AuthVerifier` implementations for the most common situations. These include situations where remote clients use HTTP Basic or HTTP Digest authentication, send credentials in request parameters, send authenticated `JSESSIONID`s, or use shared secrets to establish trust. Other `AuthVerifier` implementations can be deployed as modules containing implementations of the `AuthVerifier` interface that are registered as services in Liferay DXP's OSGi runtime.

Note: The authentication verification layer's focus is on verifying authentication, not on providing credentials. The authentication verification layer is NOT responsible for issuing tokens, credentials, or displaying Sign In portlets. Instead, the layer verifies existing credentials and authenticated sessions and is therefore a complement to authentication endpoints. However, to ensure backwards compatibility, the default portal implementations support requests providing username and password credentials. Thus, the authentication verification layer stands on the border between authentication and authorization.

Authentication Verification Process Overview

This layer and surrounding processes are provided by the `AuthVerifierFilter` class that implements the `javax.servlet.Filter` interface.

Step 1: Verify Request Credentials

The layer uses the chain of responsibility design pattern to support both built-in and third party `AuthVerifier` implementations. Each `AuthVerifier` can provide configurations where it specifies mapped URLs and other properties.

Each incoming request is matched against all registered `AuthVerifiers` to select the final list of `AuthVerifiers` that is used to process the request. It's the responsibility of each `AuthVerifier` to verify the incoming request credentials.

Step 2: Create an Authorization Context

When a request is processed by all matching `AuthVerifiers`, Liferay DXP creates an authorization context for the resolved user.

This encompasses setting the `HttpServletRequest` `remoteUser` to return the resolved user ID setting Liferay DXP `ThreadLocals` to the resolved user.

The resolved user can be the user returned by one of the `AuthVerifier` instances or a guest user if no instance was able to verify the provided credentials.

For more detailed technical information, please see the AuthVerifiers (not yet written) tutorial.

Related Topics

(Coming Soon)

41.11 LDAP

Liferay DXP fully supports LDAP as a user store. Use the LDAP tab in Instance Settings's Authentication page to connect Liferay DXP to an LDAP directory. Users can be imported into Liferay DXP from LDAP or exported to LDAP from Liferay DXP. If your organization already stores user information on an LDAP server, it's convenient for both users and administrators to simply have the LDAP user information imported into Liferay DXP. Importing LDAP user information to Liferay DXP means that users don't have to remember an extra set of credentials for Liferay DXP. Importing LDAP user information to Liferay DXP also means that administrators don't have to create a whole new set of user accounts for Liferay DXP. In this article, you'll learn how to connect Liferay DXP to an LDAP server and how to configure import settings, export settings, and related LDAP configuration settings.

Configuring Liferay DXP's LDAP Settings

To access Liferay DXP's LDAP configuration settings, navigate to *Control Panel → Configuration → Instance Settings*, then scroll down and expand the form's *Authentication* section. Go to the *LDAP* tab. Use this form to connect Liferay DXP to an LDAP directory.

You configure the global values from the LDAP tab of the Authentication page.

Enabled: Check this box to enable LDAP Authentication.

Required: Check this box if LDAP authentication is required. Liferay DXP then won't allow a user to log in unless he or she can successfully bind to the LDAP directory first. Uncheck this box if users with Liferay DXP accounts but no LDAP accounts can log in to Liferay DXP.

LDAP Servers: Liferay DXP supports connections to multiple LDAP servers. Use the Add button beneath this heading to add LDAP servers. Each LDAP server has the following configuration options:

Import/Export: You can import and export user data from LDAP directories using the following options:

- *Enable Import:* Checking this box to cause Liferay DXP to do a mass import from your LDAP directories. Leave this unchecked to keep the default behavior, which synchronizes users only when they log in. Definitely leave this unchecked if you are working in a clustered environment. Otherwise, all of your nodes would try to do a mass import when each of them starts up.
- *Enable Export:* Check this box to enable Liferay DXP to export user accounts from its database to LDAP. Liferay DXP uses a listener to track any changes made to the `User` object. Liferay DXP pushes updates

out to the LDAP server whenever a `User` object is modified. Note that by default on every login, fields such as `lastLoginDate` are updated. When export is enabled, this has the effect of causing a user export every time the user logs in. You can prevent updates to users' `lastLoginDate` fields from triggering LDAP user exports by setting the following property in your `portal-ext.properties` file:

```
users.update.last.login=false
```

- *Enable Import on Startup:* Checking this box instructs Liferay DXP to run the LDAP user import when it starts up. Note: This box only appears if you check the *Enable Import* box described above.

Use LDAP Password Policy: Liferay DXP uses its own password policy by default. This can be configured on the Control Panel's Password Policies page. Check the *Use LDAP Password Policy* box if you want to use the password policies defined by your LDAP directory. Once this is enabled, the Password Policies tab states that you are not using a local password policy. You must now use your LDAP directory's mechanism for setting password policies. Liferay DXP cannot enforce these policies; the best it can do is pass through the messages returned by your LDAP server. It does this by parsing the messages in the LDAP controls the server returns. By default, Liferay DXP is configured to parse the messages returned by the Fedora Directory Server. If you use a different LDAP server, you must customize the messages in *System Settings → Foundation → System LDAP Configuration*.

Once you've finished configuring LDAP, click the *Save* button.

LDAP Options Available in System Settings

Although most LDAP configuration can be done from Instance Settings, there are several configuration parameters that are only available in System Settings. In previous versions of Liferay DXP, system scoped settings for LDAP were set in the `portal.properties` file and modified using a `portal-ext.properties` file. Those settings must now be made via System Settings.

If you need to change any of these options, navigate to *Control Panel → Configuration → System Settings*. Go to the *Foundation* section and find the entries with LDAP in the title.

Note: To use `config` files for LDAP server configuration, you must specify the Virtual Instance ID (in the source, the variable name is `companyId`) in the exported configuration file, because servers are defined at the instance scope, not the system scope. To do this, specify the virtual instance ID somewhere in the file like this:

```
companyId=1234
```

You can find your Virtual Instance ID in Control Panel → Configuration → Virtual Instances.

- On the *LDAP Auth* page, you can set the authentication method and the password encryption algorithm. The Bind authentication method is preferred by most vendors so you don't have to worry about encryption strategies. Password compare does exactly what it sounds like: it reads the user's password out of LDAP, decrypts it and compares it with the user's password in Liferay DXP, syncing the two. If you use password compare, you can also choose the encryption algorithm to use for the comparison.

- On the *LDAP Import* page, you can configure import settings from LDAP. One example is the import methods. If you set this to User, Liferay DXP imports all users from the specified portion of the LDAP tree. If you set this to Group, Liferay DXP searches all the groups and imports the users in each group. If you have users who do not belong to any groups, they are not imported.

- Use the *System LDAP Configuration* entry to manage error properties like *Error password age keywords* which lets you set a list of phrases from error messages which can possibly be returned by the LDAP server. When a user binds to LDAP, the server returns *controls* with its response of success or failure. These controls contain a message describing the error or the information that is returned with the response. Though the controls are the same across LDAP servers, the messages can be different. The properties described here contain snippets of words from those messages and work with Red Hat's Fedora Directory Server. If you are not using that server, the word snippets may not work with your LDAP server. If they don't, you can replace the values of these properties with phrases from your server's error messages. This enables Liferay DXP to recognize them.

Note: When you make a change in System Settings, it takes effect for the virtual instance you're in. If after changing a setting you create a new virtual instance, that virtual instance inherits the settings of the one it was created from as defaults. For example, say you have virtual instances named A, B, and C. From A, you modify *Error password history keywords*. This change appears only in A, not in B or C. Then from A, you create virtual instance D. The change to *Error password history keywords* appears in D (not B or C), since D defaults to A's settings because you created it from A.

In summary, if there's a configuration you need to set up Liferay DXP with LDAP, and you don't find it in Instance Settings, look in the LDAP System Settings entries.

Adding LDAP Servers

Click on the *Add* button beneath the LDAP Servers heading to add an LDAP server connection. If you have more than one LDAP server, you can arrange the servers by order of preference using the up/down arrows. When you add an LDAP Server, you must provide several pieces of data so Liferay DXP can bind to that LDAP server and search it for user records. Regardless of how many LDAP servers you add, each server has the same configuration options.

Server Name: Enter a name for your LDAP server.

Default Values: Several leading directory servers are listed here. If you are using one of these, select it and click the *Reset Values* button. The rest of the form will be populated with the proper default values for that directory.

Connection: These settings cover the basic connection to LDAP.

- *Base Provider URL:* The link to the LDAP server. Make sure the Liferay DXP server can communicate with the LDAP server. If there is a firewall between the two systems, check to make sure the appropriate ports are opened.
- *Base DN:* The Base Distinguished Name for your LDAP directory. It is usually modeled after your organization. For a commercial organization, it may look similar to this: `dc=companynamehere,dc=com`.
- *Principal:* By default, the LDAP administrator user ID is populated here. If you have removed the default LDAP administrator, you will need to use the fully qualified name of the administrative credential that you use instead. You need an administrative credential because Liferay DXP uses this ID to synchronize user accounts to and from LDAP.
- *Credentials:* This is the password for the LDAP administrative user.

This is all you need to make a regular connection to an LDAP directory. The rest of the configuration is optional. The default attribute mappings usually provide enough data to synchronize back to the Liferay

DXP database when a user attempts to log in. To test the connection to your LDAP server, click the *Test LDAP Connection* button.

Checkpoint

Before proceeding to fine tune Liferay DXP's LDAP connections, ensure the following steps have been taken:

1. The LDAP connection has been enabled in the *Control Panel*. Depending on your needs, LDAP authentication may be required so that only users who have been bound may log in.
2. *Export/Import*: for users in a clustered environment, this should be disabled so that there are no massive imports on every node upon start up.
3. When adding the LDAP server, the *Server Name*, *Default Values*, *Connection* values are correct. It is always a good idea to click the *Test LDAP Connection* before saving.

Security

If you are running your LDAP directory in SSL mode to prevent credential information from passing through the network unencrypted, you must perform extra steps to share the encryption key and certificate between the two systems.

For example, if your LDAP directory is Microsoft Active Directory on Windows Server 2003, you'd share the certificate like this:

Click *Start* → *Administrative Tools* → *Certificate Authority*. Highlight the machine that is the certificate authority, right-click on it, and click *Properties*. From the General menu, click *View Certificate*. Select the Details view, and click *Copy To File*. Use the resulting wizard to save the certificate as a file. You will also need to import the certificate into the *cacerts keystore*. The import is handled by a command like the following:

```
keytool -import -trustcacerts -keystore /some/path/jdk1.5.0_11/jre/lib/security/cacerts -storepass changeit -noprompt -alias MyRootCA -
file /some/path/MyRootCA.cer
```

The *keytool* utility ships as part of the Java SDK.

Once this is done, go back to the LDAP page in the Control Panel. Modify the LDAP URL in the Base DN field to the secure version by changing the protocol to `ldaps` and the port to 636 like this:

```
ldaps://myLdapServerHostname:636
```

Save the changes. Your Liferay DXP now encrypts its authentication to LDAP.

Managing LDAP Server

Users: This section contains settings for finding users in your LDAP directory.

- *Authentication Search Filter:* The search filter box can be used to determine the search criteria for user logins. By default, Liferay DXP uses users' email addresses for their login names. If you have changed this setting, you must modify the search filter here, which has been configured to use the email address attribute from LDAP as a search criterion. For example, if you changed Liferay DXP's authentication method to use screen names instead of the email addresses, you would modify the search filter so it can match the entered log in name:

    ```
    (cn=@screen_name@)
    ```

- *Import Search Filter:* Depending on the **LDAP** server, there are different ways to identify the user. The default setting is usually fine:

  ```
  (objectClass=inetOrgPerson)
  ```

  If you want to search for only a subset of users or users that have different LDAP object classes, you can change this.

- *User Mapping:* The next series of fields allows you to define mappings from LDAP attributes to Liferay DXP fields. Though your LDAP user attributes may be different from LDAP server to LDAP server, there are five fields Liferay DXP requires to be mapped for the user to be recognized:

  - *Screen Name* (e.g., *uid*)
  - *Password* (e.g., *userPassword*)
  - *Email Address* (e.g., *mail* or *email*)
  - *First Name* (e.g., *name* or *givenName*)
  - *Last Name* (e.g., *sn*)

Note: If you intend to create or import users with no email addresses, then you must set `users.email.address.required=false` in your `portal-ext.properties`. With this set, Liferay auto-generates an email address combining the user ID plus the suffix defined in the property `users.email.address.auto.suffix=`. Finally, make sure to set Liferay and LDAP authentication to something other than email address.

```
If you want to import LDAP groups as Liferay DXP user groups, make sure to
define a mapping for the Liferay DXP group field so that membership information
is preserved:

+   *Group* (e.g., *member*)

The other LDAP user mapping fields are optional.
```

The Control Panel provides default mappings for commonly used LDAP attributes. You can also add your own mappings.

- *Test LDAP Users:* Once you have your attribute mappings set up (see above), click the *Test LDAP Users* button and Liferay DXP will attempt to pull LDAP users and match them with their mappings as a preview.

Groups: This section contains settings for mapping LDAP groups to Liferay DXP user groups.

- *Import Search Filter:* This is the filter for finding the LDAP groups that you want to map to Liferay DXP user groups. E.g.,

  ```
  (objectClass=groupOfNames)
  ```

  Enter the LDAP group attributes you want retrieved for this mapping. The following attributes can be mapped. The *Group Name* and *User* fields are required, the *Description* is optional.

  - *Group Name* (e.g., *cn* or *o*)
  - *Description* (e.g., *description*)

Figure 41.9: Testing LDAP Users

  - *User* (e.g., *member*)

- *Test LDAP Groups:* Click the *Test LDAP Groups* button to display a list of the groups returned by your search filter.

Export: This section contains settings for exporting user data from LDAP.

- *Users DN:* Enter the location in your LDAP tree where the users will be stored. When Liferay DXP does an export, it will export the users to this location.
- *User Default Object Classes:* When a user is exported, the user is created with the listed default object classes. To find out what your default object classes are, use an LDAP browser tool such as JXplorer to locate a user and view the Object Class attributes stored in LDAP for that user.
- *Groups DN:* Enter the location in your LDAP tree where the groups will be stored. When Liferay DXP does an export, it exports the groups to this location.
- *Group Default Object Classes:* When a group is exported, the group is created with the listed default object classes. To find out what your default object classes are, use an LDAP browser tool such as *Jxplorer* to locate a group and view the Object Class attributes stored in LDAP for that group.

Once you set all your options and tested your connection, click *Save*. From here, you can add another LDAP server or set just a few more options that apply to all of your LDAP server connections.

Note: If a user changes a value like a password in Liferay DXP, that change is passed to the LDAP server, provided Liferay DXP has enough schema access to make the change.

Now you know how to connect an LDAP server to Liferay DXP and how to configure user import behavior, export behavior, and other LDAP settings.

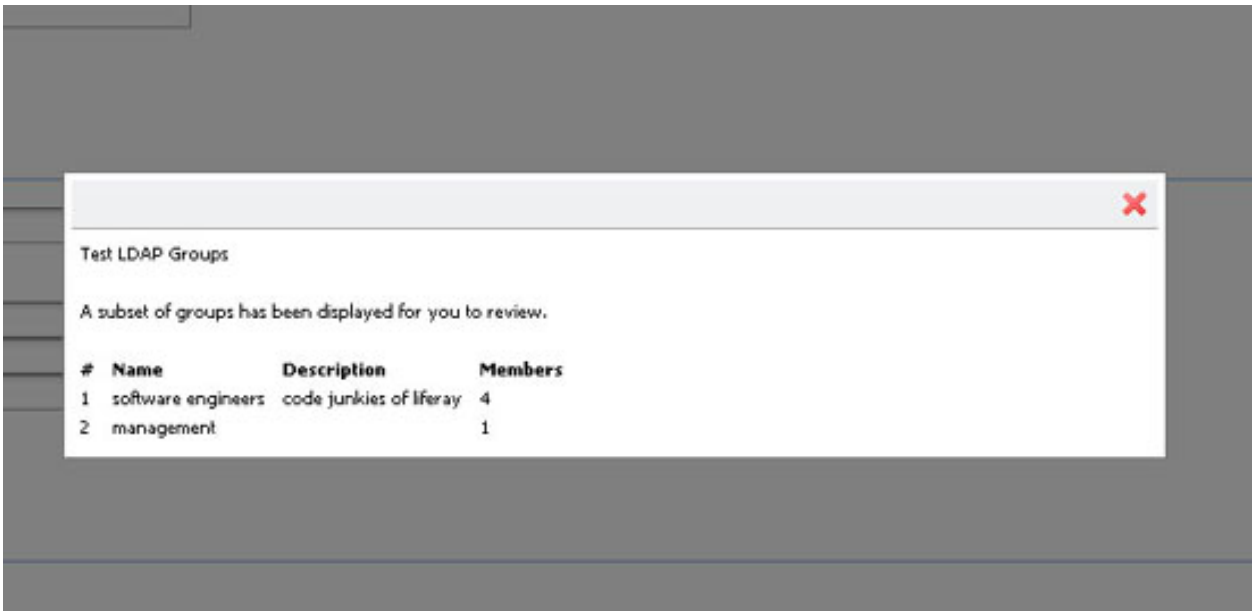


Figure 41.10: Mapping LDAP Groups

Related Topics

Liferay DXP Security Overview Logging into Liferay DXP

41.12 Token-based Single Sign On Authentication

Token-based SSO authentication was introduced in 7.0 to standardize support for Shibboleth, SiteMinder, Oracle OAM, or any other SSO product that works by propagating a token via one of the following mechanisms:

- HTTP request parameter
- HTTP request header
- HTTP cookie
- Session attribute

Since these providers have a built-in web server module, you should use the Token SSO configuration.

The authentication token contains either the Liferay DXP user's screen name or email address, whichever Liferay DXP has been configured to use for the particular company (portal instance). Recall that Liferay DXP supports three authentication methods:

- By email address
- By screen name
- By user ID

Note that Liferay DXP's token-based authentication mechanism only supports email address and screen name. If the portal is configured to use user ID when a token-based authentication is attempted, the `TokenAutoLogin` class logs this warning:

```
Incompatible setting for: company.security.auth.type
```

Please note that the above sources are fully trusted.

Furthermore, you must use a security mechanism external to Liferay DXP, such as a fronting web server like Apache. The chosen fronting solution must prevent malicious Liferay DXP user impersonation that otherwise might be possible by sending HTTP requests directly to Liferay DXP from the client's web browser.

Token based authentication is disabled by default. To manage token based SSO authentication, navigate to Liferay DXP's Control Panel, click on *System Settings*, then click *Foundation*. The Token Based SSO is located on page 3. Alternately, you can search for *Token* in the Search field. Here are the configuration options for the Token Based SSO module:

Authentication cookies: Set this to the cookie names that must be removed after logout. (Example: `SMIDENTITY`, `SMSESSION`)

Enabled: Check this box to enable token-based SSO authentication.

Import from LDAP: Check this box to automatically import users from LDAP if they do not exist in the portal.

Logout redirect URL: When user logs out of Liferay DXP, the user is redirected to this URL.

Token location: Set this to the location of the user token. As mentioned earlier, the options are:

- HTTP request parameter
- HTTP request header
- HTTP cookie
- Session attribute

User token name: Set equal to the name of the token. This will be retrieved from the specified location. (Example: SM_USER)

Remember to click *Save* to activate Token Based SSO.

Required SiteMinder Configuration

If you use SiteMinder, note that Liferay DXP sometimes uses the tilde character in its URLs. By default, SiteMinder treats the tilde character (and others) as bad characters and returns an HTTP 500 error if it processes a URL containing any of them. To avoid this issue, change this default setting in the SiteMinder configuration to this one:

```
BadUrlChars       //,./,/.,/*,*.,\,%00-%1f,%7f-%ff,%25
```

The configuration above is the same as the default except the ~ was removed from the bad URL character list. Restart SiteMinder to make your configuration update take effect. For more information, please refer to SiteMinder's documentation

Summary

Liferay DXP's token-based SSO authentication mechanism is highly flexible and compatible with any SSO solution which can provide it with a valid Liferay DXP user's screen name or email address. These include Shibboleth and SiteMinder.

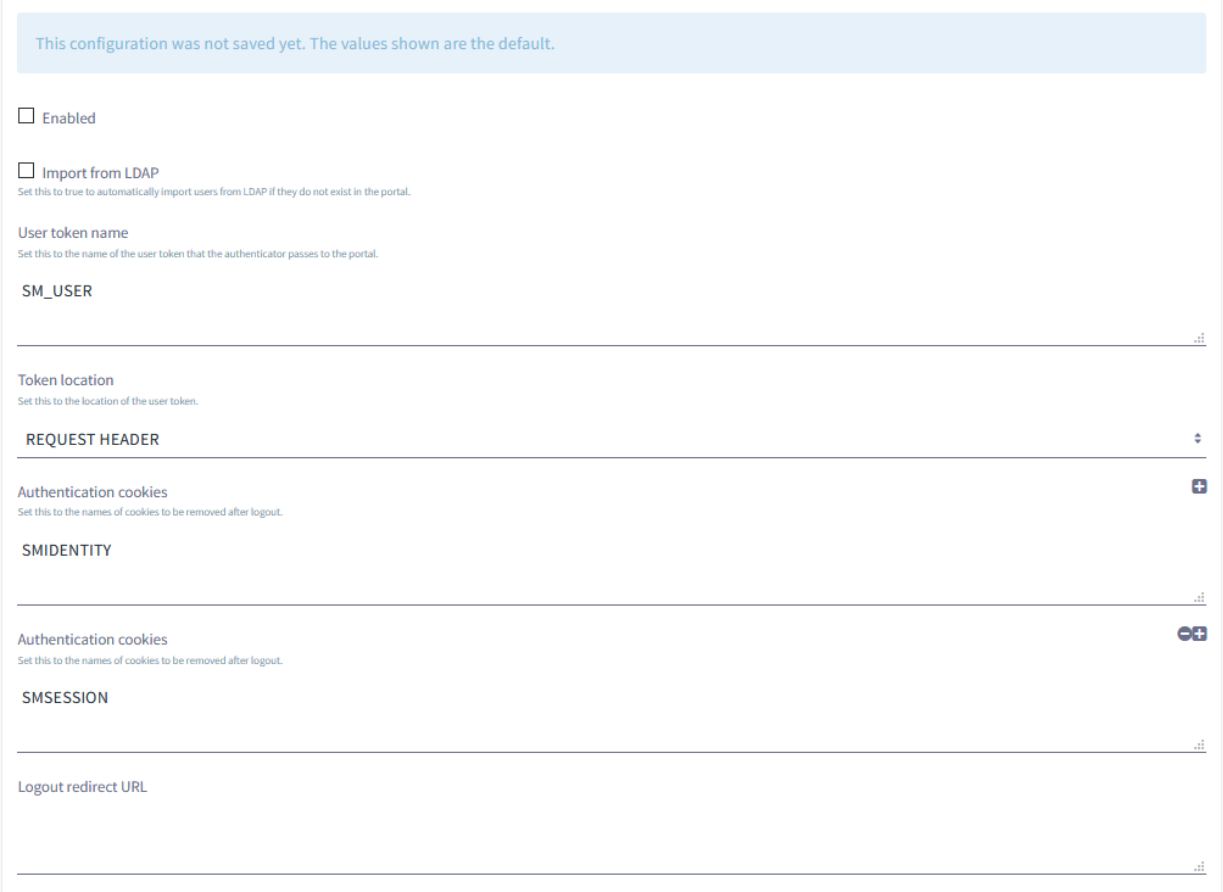


Figure 41.11: The form in the Control Panel provides a straightforward way to configure Token Based SSO.

41.13 Authenticating with OpenID Connect

Note: OpenID Connect authentication is available in Liferay DXP on Fix Pack 79 or higher patch level.

OpenID Connect is a lightweight authentication layer built on top of the OAuth 2.0 authorization protocol. It compliments having local accounts by enabling users to authenticate using accounts they have on other systems. Users who avoid signing up for new accounts can then use an account they already have to sign into your website. By using OpenID Connect, you *delegate* user authentication to other providers, making it easy for users with existing accounts to authenticate to your system.

Note: You can add multiple providers to your installation, but Liferay DXP can't yet be an OpenID Connect provider.

Because OpenID Connect is built on OAuth 2.0, its token flow is similar. OAuth 2.0 is only an authorization protocol, so it sends an *access token* that grants access to particular APIs. OpenID Connect adds to this an *identity token* that passes user information like name and email, provided the user has authenticated and granted permission.

Creating a Client in OpenID Connect Provider

To use OpenID Connect, you must first register it as a client in your provider. This is an OAuth 2.0 client. The process varies by provider:

1. Navigate to the provider's website and create a client.

2. During the creation process, you must supply an *authorized redirect URL* that can process the tokens sent from the provider. Liferay DXP's URL is

   ```
   https://[server.domain]/c/portal/login/openidconnect
   ```

3. The provider will send several pieces of information. Some of these, like the Discovery Endpoint, Authorization Endpoint, or Issuer URL are the same regardless of the client. The two pieces of information unique to your request are the `client_id` and the `client_secret`.

Collect the information from the provider. You'll need it create the provider next.

Configuring an OpenID Connect Provider Connection

Go to *Control Panel* → *Configuration* → *System Settings* → *Foundation* and select ***OpenID Connect Provider*** (*System Scope*) and follow these steps:

1. Add the provider by clicking the *Add* button.

2. Use the information you received from the provider to fill out the form:

Provider Name: This name appears in the Sign-In Portlet when users use OpenID Connect to log in.
OpenID Client ID: Provide the OAuth 2.0 Client ID you received from your provider.
OpenID Connect Client Secret: Provide the OAuth 2.0 Client Secret you received from your provider.
Scopes: Leave the default, which requests the user name and the email. Your provider may offer other scopes of user information.
Discovery Endpoint: Other URLs may be obtained from this URL, and they vary by provider.
Authorization Endpoint: This URL points to the provider's URL for authorizing the user (i.e., signing the user in).
Issuer URL: The provider's URL that points to information about the provider who is issuing the user information.
JWKS URI: A URL that points to the provider's JSON Web Key Set that contains the public keys that can verify the provider's tokens.
ID Token Signing Algorithms: Set the supported ID token algorithms manually. Normally, this is "discovered" at the discovery endpoint. You can add as many of these as you need.
Subject Types: A Subject Identifier is a unique and never reassigned identifier the provider uses to establish who the user is, and is consumed by the client (i.e., Liferay DXP). There are two types: public (provides the same value to all clients) and private (provides a different value to each client).
Token Endpoint: The provider's URL where tokens can be requested.
User Information Endpoint: The OAuth 2.0 protected URL from which user information can be obtained.
Once you've filled out the form, click *Save*, and you're ready to enable OpenID Connect authentication.

System Settings configuration file:

```
com.liferay.portal.security.sso.openid.connect.internal.configuration.OpenIdConnectProviderConfiguration-[name].config
```

where [name] is a descriptive, but unique name for example provider1.

Enabling OpenID Connect Authentication

1. Go to *Control Panel* → *Configuration* → *System Settings* → *Foundation* and select ***OpenID Connect***.
2. Click the *Enabled* check box, and then click *Save*.

Note: You can also enable OpenID Connect authentication for the given virtual instance through the *Control Panel* → *Configuration* → *Instance Settings* → *OpenID Connect* tab.

System Settings configuration file:

```
com.liferay.portal.security.sso.openid.connect.configuration.OpenIdConnectConfiguration.config
```

Now users can sign in with OpenID Connect.

Signing In With OpenID Connect

There's a new link in the Sign-In Portlet for signing in with OpenID Connect:

1. From the Sign-In Portlet, click the OpenID Connect link at the bottom.
2. Choose a provider and click *Sign In*.
3. This takes you to your provider's sign in page. Enter your credentials and log in.
4. Upon successful authentication, you're redirected back to Liferay DXP in an authenticated state.

OpenID is a standards-based, secure way to authenticate users from other systems.

41.14 OpenID Single Sign On Authentication

OpenID is a single sign-on standard implemented by multiple vendors. Users can register for an ID with the vendor they trust. The credential issued by that vendor can be used by all the web sites that support OpenID. Some high profile OpenID vendors are Google, Paypal, Amazon, and Microsoft. Please see the OpenID site for a more complete list.

With OpenID, users don't have to register for a new account on every site which requires an account. Users register on *one* site (the OpenID provider's site) and then use those credentials to authenticate to many web sites which support OpenID. Web site owners sometimes struggle to build communities because users are reluctant to register for *another* account. Supporting OpenID removes that barrier, making it easier for site owners to build their communities. All the account information is kept with the OpenID provider, making it much easier to manage this information and keep it up to date.

Liferay DXP can act as an OpenID consumer, allowing users to automatically register and sign in with their OpenID accounts. Internally, the product uses OpenID4Java to implement the feature.

OpenID at the System Scope

OpenID is enabled by default in Liferay DXP but can be disabled or enabled at either the system scope or portal instance scope. To configure the OpenID SSO module at the system level, navigate to the Control Panel and click on *Configuration* → *System Settings*. Then click on the *Foundation* category and search for *OpenID* in the list. There's only a single configuration setting. Check the *Enabled* box to enable OpenID at the system scope (for all portal instances), uncheck it to disable it at the system scope.

OpenID at the Instance Scope

To configure the OpenID SSO module at the portal instance scope, navigate to the Control Panel and click on *Configuration* → *Instance Settings*, then on *Authentication* → *OpenID*. There's only a single configuration setting. Check the *Enabled* box to enable OpenID for the current portal instance, or uncheck it to disable it for the current portal instance.

Regardless of whether OpenID is enabled at the System or Instance scope, users can see the OpenID icon when they sign into Liferay DXP. Click *Sign In*. The OpenID icon is displayed at the lower left.

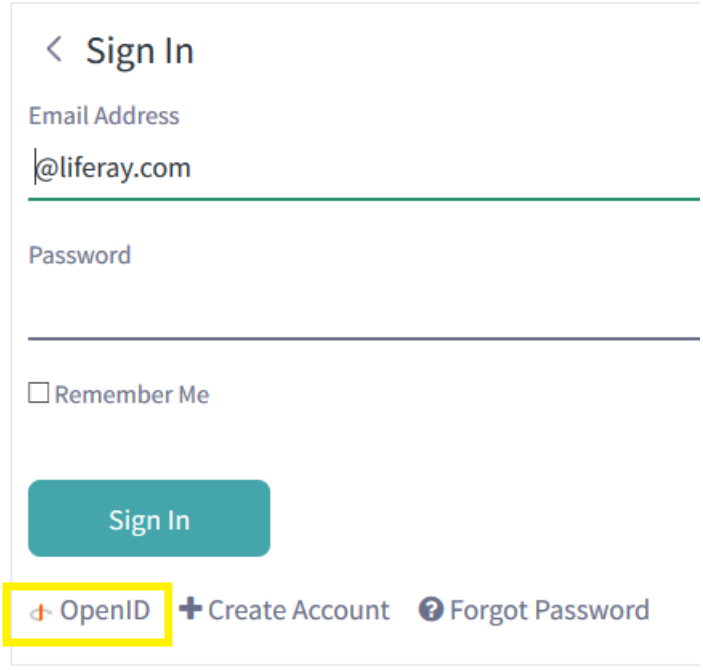


Figure 41.12: The OpenID icon is at the bottom of the Sign In Portlet

Related Topics

Liferay DXP Security Overview Token-based Single Sign On Authentication CAS Single Sign On Authentication OpenAM Single Sign On Authentication

41.15 CAS (Central Authentication Service) Single Sign On Authentication

CAS is an authentication system originally created at Yale University. It is a widely used open source single sign-on solution and was the first SSO product to be supported by Liferay DXP. Liferay DXP's CAS module includes the CAS client, so there's no need to install it separately.

Note: Liferay DXP supports CAS 3.3.x.

The CAS Server application requires your server to have a properly configured Secure Socket Layer (SSL) certificate. To generate one yourself, use the `keytool` utility that comes with the JDK. First generate the key, then export the key into a file. Finally, import the key into your local Java key store. For public, internet-based production environments, you will need to either purchase a signed key from a recognized certificate authority or have your key signed by a recognized certificate authority. For intranets, you should have your IT department pre-configure users' browsers to accept the certificate so they don't get warning messages about the certificate.

To generate a key, use the following command:

```
keytool -genkey -alias tomcat -keypass changeit -keyalg RSA
```

Instead of the password in the example (`changeit`), use a password you will remember. If you are not using Tomcat, you may want to use a different alias as well. For first and last names, enter `localhost` or the host name of your server. It cannot be an IP address.

To export the key to a file, use the following command:

```
keytool -export -alias tomcat -keypass changeit -file server.cert
```

Finally, to import the key into your Java key store, use the following command:

```
keytool -import -alias tomcat -file server.cert -keypass changeit -keystore $JAVA_HOME/jre/lib/security/cacerts
```

If you are on a Windows system, replace `$JAVA_HOME` above with `%JAVA_HOME%`. Of course, all of this needs to be done on the system where CAS is running.

Once your CAS server is up and running, configure Liferay DXP to use it. CAS configuration can be applied either at the system scope or at the scope of a portal instance. To configure the CAS SSO module at the system scope, navigate to the Control Panel, click on *Configuration → System Settings*, click on the *Foundation* category, and find the CAS Module entry. The values configured there provide the default values for all portal instances. Enable CAS authentication and then modify the URL properties to point to your CAS server.

Enabled: Check this box to enable CAS single sign-on.

Import from LDAP: A user may be authenticated from CAS and not yet exist in Liferay DXP. Select this to automatically import users from LDAP if they do not exist in Liferay DXP. For this to work, LDAP must be enabled.

The rest of the settings are various URLs, with defaults included. Change *localhost* in the default values to point to your CAS server. When you are finished, click *Save*. After this, when users click the *Sign In* link, they will be directed to the CAS server to sign in to Liferay DXP.

For some situations, it might be more convenient to specify the system configuration via files on the disk. To do so, simply create the following file:

```
{LIFERAY_HOME}/osgi/configs/com.liferay.portal.security.sso.cas.module.configuration.CASConfiguration.cfg
```

The format of this file is the same as any properties file. The key to use for each property that can be configured is shown below. Enter values in the same format as you would when initializing a Java primitive type with a literal value.

Property Label | Property Key | Description | Type **Enabled** | `enabled` | Check this box to enable CAS SSO authentication. | `boolean` **Import from LDAP** | `importFromLDAP` | Users authenticated from CAS that do not exist in Liferay DXP are imported from LDAP. LDAP must be enabled separately. | `boolean` **Login URL** | `loginURL` | Set the CAS server login URL. | `String` **Logout on session expiration** | `logoutOnSessionExpiration` | If checked, browsers with expired sessions are redirected to the CAS logout URL. | `boolean` **Logout URL** | `logoutURL` | The CAS server logout URL. Set this if you want Liferay DXP's logout function to trigger a CAS logout | `String` **Server Name** | `serverName` | The name of the Liferay DXP instance (e.g., `liferay.com`). If the provided name includes the protocol (`https://`, for example) then this will be used together with the path `/c/portal/login` to construct the URL to which the CAS server will provide tickets. If no scheme is provided, the scheme normally used to access the Liferay DXP login page will be used. | `String` **Server URL** | `serviceURL` | If provided, this will be used as the URL to which the CAS server provides tickets. This overrides any URL constructed based on the Server Name as above. | `String` **No Such User Redirect URL** | `noSuchUserRedirectURL` | Set the URL to which to redirect the user if the user can authenticate with CAS but cannot be found in Liferay DXP. If import from LDAP is enabled, the user is redirected if the user could not be found or could not be imported from LDAP. | `String`

To override system defaults for a particular portal instance, navigate to the Control Panel, click on *Configuration → Instance Settings*, click on *Authentication* on the right and then on *CAS* at the top.

Summary

System administrators sometimes think it's sufficient to link multiple webapps to a single user directory such as LDAP. This certainly goes a long way towards providing a convenient user experience. However, if this approach is taken, the overall security of your users' credentials is reduced to that of the weakest webapp. Thus, this approach should only be taken if you're extremely confident in the security of each linked webapp. If you're not, you should consider an alternative approach to authentication.

CAS authentication is very straight-forward to set up. Using it removes the risk associated with asking users to enter their credentials into multiple web applications.

41.16 OpenAM Single Sign On Authentication

OpenAM is an open source single sign-on solution that comes from the code base of Sun's System Access Manager product. Liferay DXP integrates with OpenAM, allowing you to use OpenAM to integrate Liferay DXP into an infrastructure that contains a multitude of different authentication schemes against different repositories of identities.

Note that OpenAM relies on cookie sharing between applications. Thus, in order for OpenAM to work, all applications that require SSO must be in the same web domain.

You can set up OpenAM on the same or different server as Liferay DXP . If you are using the same Liferay DXP server to host your OpenAM, you must deploy the OpenAM `.war`. The `.war` is available here. Otherwise, follow the instructions at the OpenAM site to install OpenAM. Once you have it installed, create the Liferay DXP administrative user in it. Users are mapped back and forth by screen names. By default, the Liferay DXP administrative user has a screen name of *test*, so if you were to use that account, register the user in OpenAM with the ID of *test* and the email specified in the `admin.email.from.address` portal property. Once you have the user set up, log in to OpenAM using this user.

In the same browser window, log in to Liferay DXP as the administrative user (using the admin email address specified previously). Go to the Control Panel and click *Configuration → Instance Settings → Authentication*

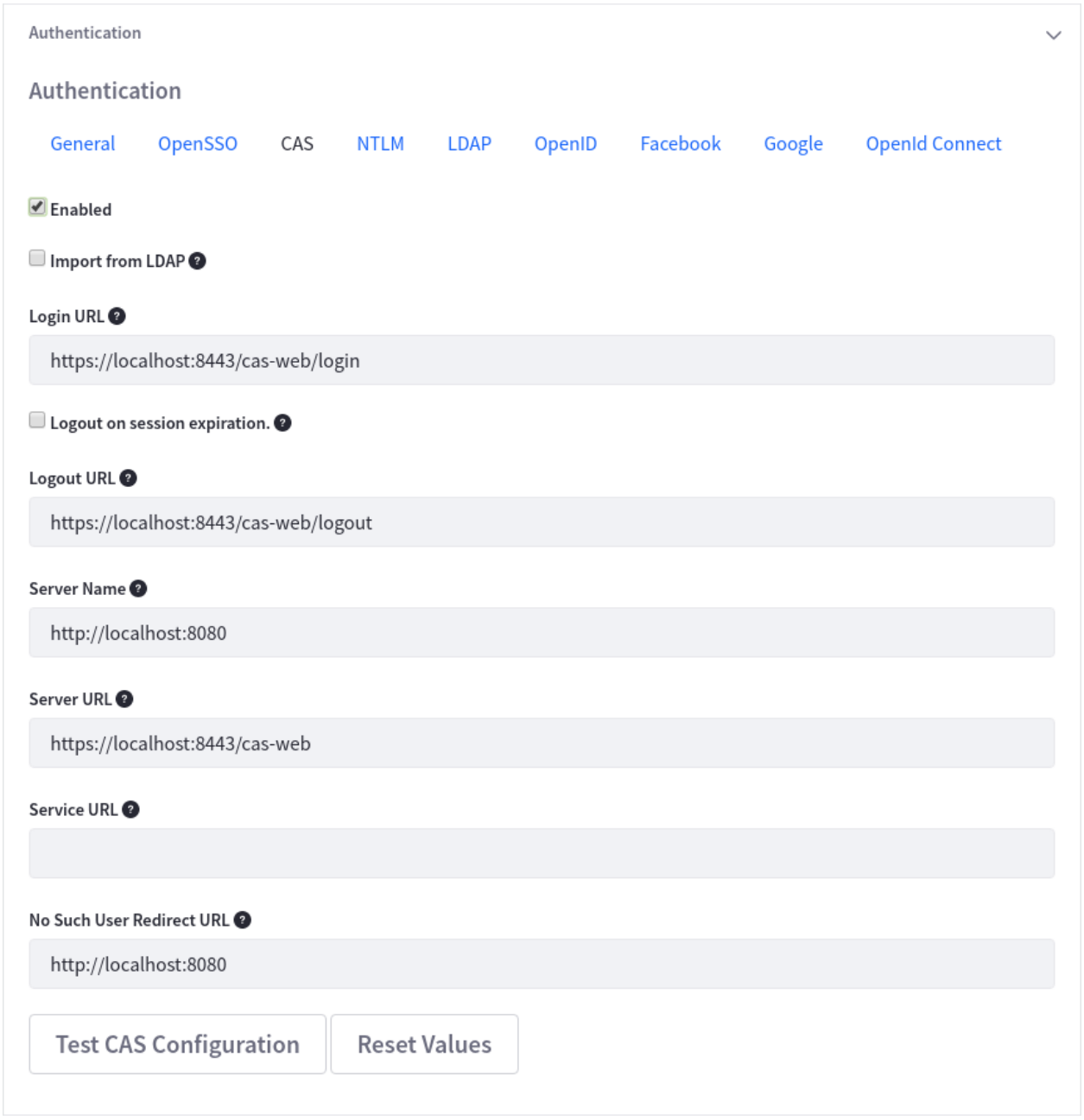


Figure 41.13: shows the CAS tab on the Instance Setting's Authentication section before configuration.

→ *OpenSSO* at the top. Modify the three URL fields (Login URL, Logout URL, and Service URL) so they point to your OpenAM server (in other words, only modify the host name portion of the URLs), check the *Enabled* box, and click *Save*. Liferay DXP then redirects users to OpenAM when they request the `/c/portal/login` URL *for example, when they click on the* Sign In* link).

Liferay DXP's OpenAM configuration can be applied at either the system scope or at the instance scope. To configure the OpenAM SSO module at the system scope, navigate to Liferay DXP's Control Panel, click on *Configuration → System Settings → Foundation* and find the OpenSSO entry. Click on it and you'll find these settings to configure. The values configured here provide the default values for all portal instances. Enter the in the same format as you would when initializing a Java primitive type with a literal value.

Note: OpenAM 12 and below work with Liferay DXP, but are at end of life. Because of this, we recommend only OpenAM 13 for production use. OpenAM 13 requires Liferay DXP Fix Pack 80+ patch level.

Property Label | Property Key | Description | Type **Version** | `version` | OpenAM version to use (12 and below or 13), available in Liferay DXP Fix Pack 80+ | `String` **Enabled** | `enabled` | Check this box to enable OpenAM authentication. Note that OpenAM will work only if LDAP authentication is also enabled and Liferay DXP's authentication type is set to screen name. | `boolean` **Import from LDAP** | `importFromLDAP` | If this is checked, users authenticated from OpenAM that do not exist in Liferay DXP are imported from LDAP. LDAP must be enabled. | `boolean` **Login URL** | `loginURL` | The URL to the login page of the OpenAM server | `String` **Logout URL** | `logoutURL` | The URL to the logout page of the OpenAM server | `String` **Service URL** | `serviceURL` | The URL by which OpenAM can be accessed to use the authenticated web services. If you are using OpenAM Express 8 or higher, you need to have the server running Java 6. | `String` **Screen Name Attribute** | `screenNameAttr` | The name of the attribute on the OpenAM representing the user's screen name | `String` **Email Address Attribute** | `emailAddressAttr` | The name of the attribute on the OpenAM representing the user's email address | `String` **First Name Attribute** | `firstNameAttr` | The name of the attribute on the OpenAM representing the user's first name | `String` **Last Name Attribute** | `lastNameAttr` | The name of the attribute on the OpenAM representing the user's last name | `String`

To override these default settings for a particular portal instance, navigate to Liferay DXP's Control Panel, click on *Configuration → Instance Settings*, and then click on *Authentication* at the right and then on *OpenSSO* at the top.

Summary

OpenAM is useful when all the applications that require SSO are within the same web domain. Thus, OpenAM it is often used for intranets. OpenAM can be deployed via a fronting or reverse proxy web server such as Apache, intercepting all requests to a set of web applications. For this reason, OpenAM is also useful when implementing SSO for applications that don't support SSO out of the box.

41.17 Facebook Connect Single Sign On Authentication

Facebook Connect SSO authentication is an integration with Facebook's Graph API. It retrieves the user's Facebook profile information and matches it to existing Liferay DXP users (either by Facebook ID or by email address). Once the user's primary Facebook email address is found, the Facebook Connect SSO authentication module searches for the same email address in Liferay DXP's `User_` table. If a match is found, the user is signed on, provided that the required permissions have been granted on the Facebook side.

If a match isn't made, the user is prompted in Liferay DXP to add a user from Facebook. When new Liferay DXP users are added this way, they are created by retrieving the following four fields from Facebook:

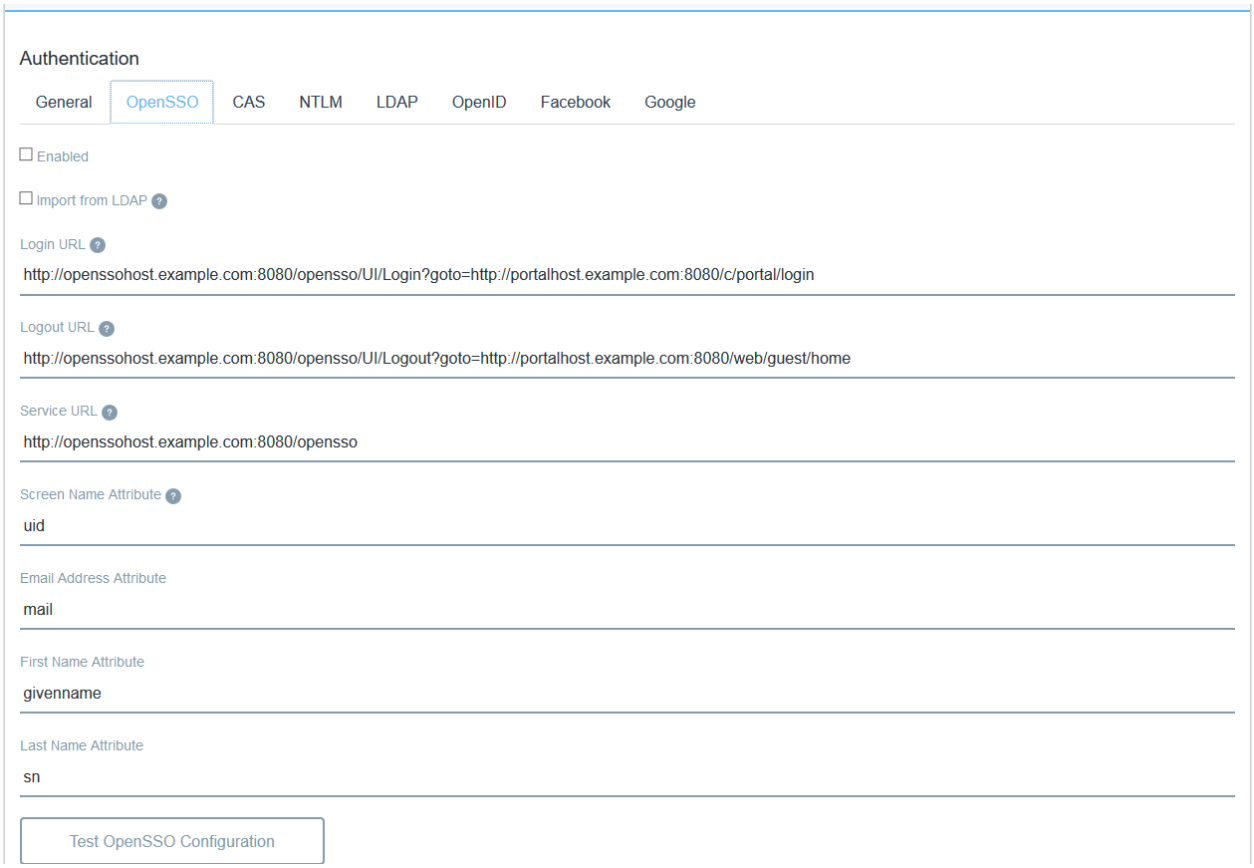


Figure 41.14: Liferay DXP's OpenSSO tab lets you configure OpanAM.

- Email address
- First name
- Last name
- Gender

In order to integrate Liferay DXP with Facebook, you must first create an "application" on Facebook's website here: https://developers.facebook.com. This is necessary because Facebook Connect requires Liferay DXP to authenticate using the OAuth 2.0 protocol. Facebook provides you the necessary application ID and secret used in OAuth messages sent between Liferay DXP and Facebook. One benefit of this is that a Facebook user can revoke Liferay DXP's access at any time.

Managing Facebook Connect SSO Authentication

Apply Facebook Connect SSO configuration at the system scope or instance scope. To configure the Facebook Connect SSO module at the System level, navigate to Liferay DXP's *Control Panel* → Configuration → System Settings, *and find the* Facebook Connect* module under the *Foundation* heading. The values configured here provide the default values for all portal instances.

To override these defaults for a particular portal instance, navigate to Liferay DXP's *Control Panel*, click on *Instance Settings*, and find *Facebook* within the Authentication section.

Enabled: Check this box to enable Facebook Connect SSO authentication.

Require Verified Account: Check this box to allow logins by Facebook users who have gone through the Facebook email verification process to prove that they can access the inbox associated with the email address they provided when registering for a Facebook account.

Application ID: This can only be set at the portal instance level. Enter the ID of your registered Facebook application.

Application Secret: This can only be set at the portal instance level. Enter the secret of your registered Facebook application.

Graph URL: This is the base URL of the Facebook graph API. Only change this if Facebook changes their graph API. If Facebook's graph API remains unchanged, use the default graph URL.

OAuth Authorization URL: This is Facebook's OAuth authorization URL. You will only need to change this if Facebook changes their OAuth authorization endpoint. This URL will be decorated with dynamic data and linked to from the Liferay Sign In portlet.

OAuth Token URL: This is Facebook's OAuth access token URL. Liferay DXP uses this URL to exchange a request token for an access token.

OAuth Redirect URL: This is the URL that the user will be directed to once an OAuth request token has been generated. The URL points to a Liferay DXP service which exchanges the request token for the access token which is required in order for Liferay DXP to make successful calls to the Facebook Graph API. You should only need to change this URL if requests to your Liferay DXP instance need to go via a fronting webserver such as Apache that does URL rewriting.

Related Topics

(Coming Soon)

41.18 NTLM Single Sign On Authentication

NTLM (NT LAN Manager) is a suite of Microsoft protocols that provide authentication, integrity, and confidentiality for users. Though Microsoft has adopted Kerberos in modern versions of Windows server, NTLM is still used when authenticating to a workgroup. Liferay DXP now supports NTLM v2 authentication. NTLM v2 is more secure and has a stronger authentication process than NTLMv1.

Note that in order to use NTLM SSO, Liferay DXP's portal instance authentication type must be set to screen name as shown here.

Note: To USE NTLM with Liferay DXP, you need to configure your browser. Consult your browser vendor's documentation for the details.

Most importantly, all users *must* be imported from an Active Directory server. NTLM (and Kerberos) works only if the users are in the AD; otherwise any SSO requests initiated by Liferay DXP will fail.

NTLM configuration can be applied either at the system scope or at the scope of a portal instance. To configure the NTLM SSO module at the system scope, navigate to the Control Panel, click on *Configuration → System Settings*, click on the *Foundation* category, and find the NTLM module. The values configured there provide the default values for all portal instances. Enter values in the same format as you would when initializing a Java primitive type with a literal value.

Property Label | Property Key | Description | Type **Enabled** | `enabled` | Check this box to enable NTLN SSO authentication. Note that NTLM will only work if Liferay DXP's authentication type is set to screen name. | `boolean` **Domain Controller** | `domainController` | Enter the IP address of your domain controller. This is the server that contains the user accounts you want to use with Liferay DXP. | `String` **Domain Controller Name** | `domainControllerName` | Specify the domain controller NetBIOS name. | `String` **Domain** |

`domain` | Enter the domain / workgroup name | `String` **Service Account** | `serviceAccount` | You need to create a service account for NTLM. This account will be a computer account, not a user account. | `String` **Service Password** | `serviceAccount` | Enter the password for the service account. | `String` **Negotiate Flags** | `negotiateFlags` | Only available at system level. Set according to the client's requested capabilities and the server's ServerCapabilities. See here. | `String`

Note the AD's name and IP address correspond to the `domainControllerName` and `domainController` settings. The `Service Account` is for the *NTLM* account (registered with NTLM), not the Liferay DXP user account.

To override system defaults for a particular portal instance, navigate to the Control Panel, click on *Configuration* → *Instance Settings*, click on *Authentication* and then on *NTLM*.

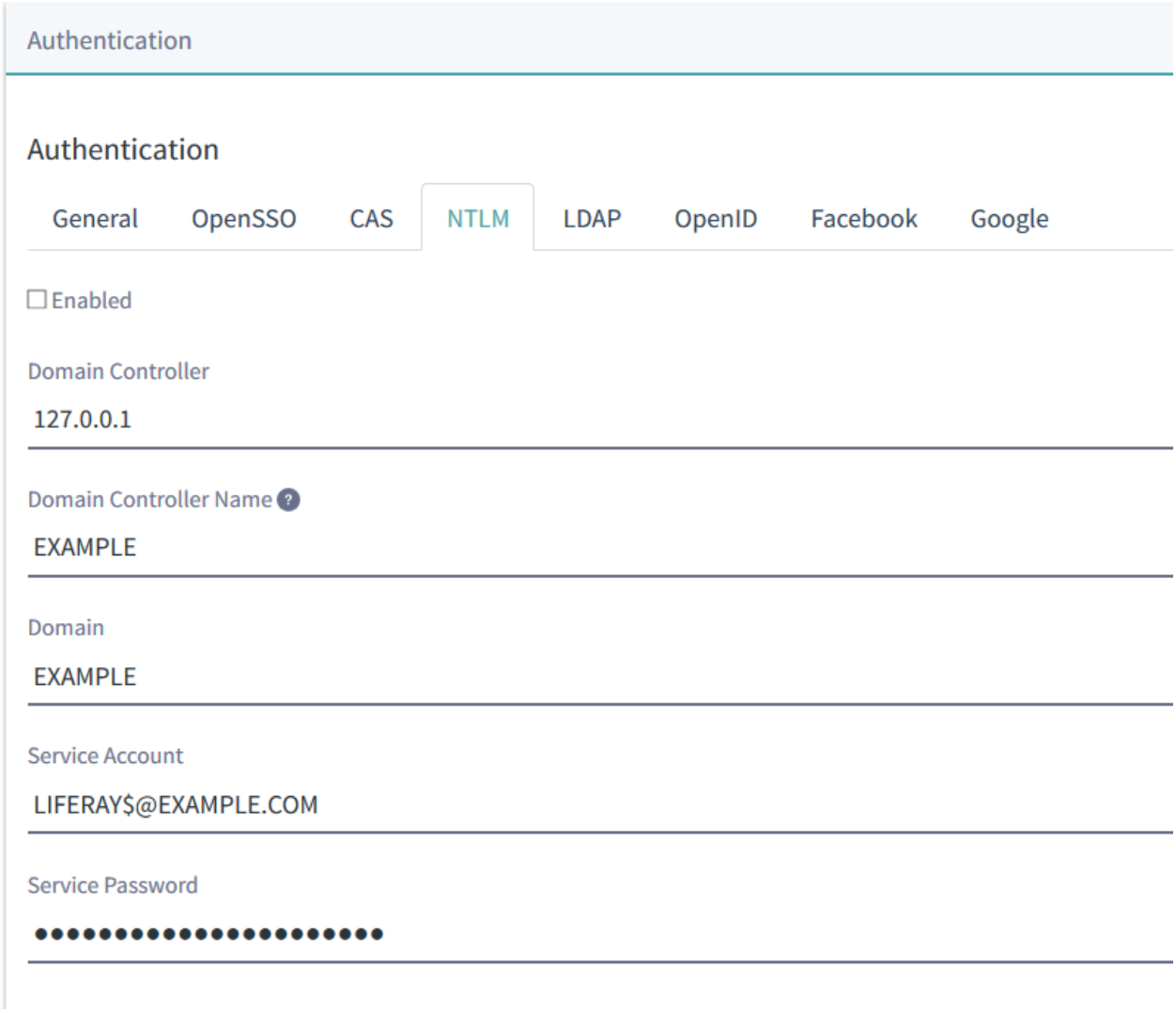


Figure 41.15: The NTLM settings on the Authentication tab let you configure SSO for Microsoft environments.

Summary

NTLM authentication is often highly desirable in Intranet scenarios where the IT department has control over what software is running on client devices and thus can ensure NTLM compatibility. In an Active Directory based network / domain, it is hard to beat the user experience that NTLM authentication can provide.

Please remember that in order to use NTLM SSO, your Liferay DXP instance authentication type must be set to screen name *and* that all users have been imported from your active directory. If this is not acceptable for your Liferay DXP implementation, then another SSO solution (such as CAS) can be used as a broker between your portal and the NTLM authentication process.

41.19 OAuth

Liferay DXP's OAuth utility authorizes third-party applications to interact with a user's resources. It's available as an app from Liferay Marketplace. Let's say you're hosting Liferay DXP and have users and customers coming to your web site. You want them to have access to a third party resource, like Twitter, and be able to access their accounts from your site. In the past, they would have to provide their Twitter user names and passwords, but not if you use OAuth. For this reason, a popular characterization for the OAuth client is the "valet key for your web services."

OAuth is a handshake mechanism where, instead of asking for personal information, Liferay DXP redirects users to a service provider like Twitter, where they can tell Twitter to allow Liferay DXP limited access to their accounts. This example is similar to our earlier "valet key" characterization. You wouldn't want a valet driver opening your glove box, storage spaces, hood, and other personal compartments in your vehicle. You would only want the valet to access what is necessary to park your car. OAuth is based on this same idea: it gives a site just enough information to do what it needs and nothing more. This assures users that their personal information is safe, but gives them freedom to take advantage of valuable resources they typically use from the service provider's site.

Registering OAuth Applications

The first thing you'll need to do is register an application for OAuth's services. To access the OAuth Admin page, navigate to the Control Panel, find *OAuth Admin*, and click on it. The OAuth Admin section will only be available if you've installed the OAuth app from Liferay Marketplace. In the OAuth Admin section, select *Add* to create a new OAuth application in the OAuth registry. You'll be given the following options:

Application Name: the display name for your application

Description: the short description that is attached to your application

Website URL: your application's URL

Callback URI: the URI where users are redirected after authentication is complete

Access Level: select the *Read* or *Write* access level. For the *Read* access level, the user can only view the application's contents, but not modify them. The *Write* access level gives the user permission to access and modify the application's contents.

After you're finished registering the OAuth app, click *Actions → View*. You'll notice Liferay DXP generated two Application Credentials: the *Consumer Key* and *Consumer Secret*. The consumer key is a value used by the application to identify itself to the service provider. Likewise, the consumer secret is a value the application uses to establish ownership of the consumer key.

Take note of your application credentials; you'll need them when configuring your application with OAuth. To learn how to configure an application with OAuth, visit the Authorizing Access to Services with OAuth (not yet written) tutorial. Once you have your application configured to use OAuth, visit the next section to begin authorizing requests via OAuth.

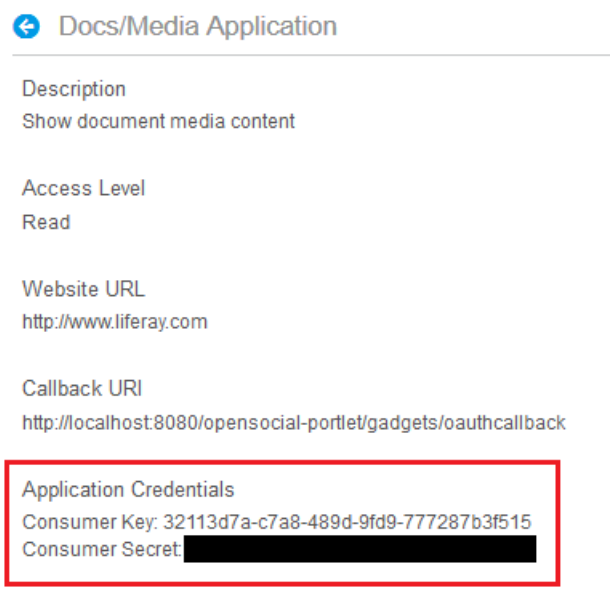


Figure 41.16: .x: You'll need the application credentials to implement OAuth in your application.

Authorizing Requests via OAuth

Once you have your application configured to use OAuth, you can place your application on a page and test out the process. Here is a basic synopsis of what's happening during the authorization process.

The app that you registered in the previous section and configured with the consumer key and secret by following the developer tutorial is characterized as a service provider. The service provider uses OAuth to allow users access to its protected resources. These protected resources are data controlled by the service provider, which can only be accessed by the application through authentication. By configuring an application in Liferay DXP to use OAuth, you're keeping all private information you have between a third-party service provider and the portal separate. Essentially, OAuth bridges the connection between the portal and third-party services without the user sharing any protected resources between them.

Once your OAuth application is placed on a portal page, you'll be asked to grant or deny the third-party service provider limited access to your portal.

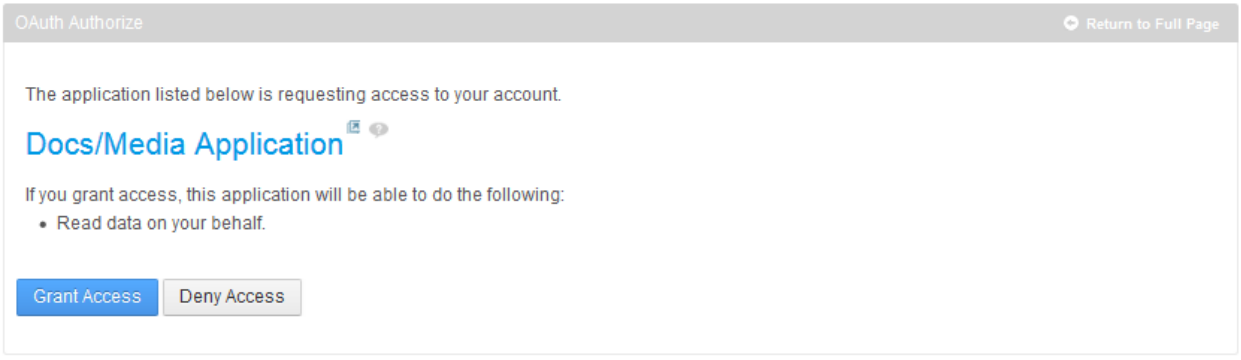


Figure 41.17: .38: You can grant or deny the service provider access to your.

Congratulations! You've successfully installed your OAuth app and authorized access between a service

provider and Liferay DXP!

41.20 AntiSamy

Liferay DXP includes an AntiSamy module that protects against malicious code that users might create. When creating content, users can include malicious code either intentionally or unintentionally. The AntiSamy module filters on specific HTML/CSS fragments and removes suspect JavaScript code from them. The module leverages the powerful OWASP AntiSamy library to enforce a content policy that's been effective for the online auction site eBay. The AntiSamy module adds an OWASP AntiSamy implementation to your portal's list of existing sanitizer implementations. Liferay DXP uses the AntiSamy sanitizer and any existing configured sanitizers to scrub user input to blogs entries, calendar events, message boards posts, wiki pages, and web content articles.

At the time of this writing, Liferay DXP's AntiSamy module uses OWASP AntiSamy 1.5.3. It's enabled by default. To configure the AntiSamy module, navigate to Liferay DXP's *Control Panel*, go to *System Settings*, open the *Foundation* category and click on *AntiSamy Sanitizer*.

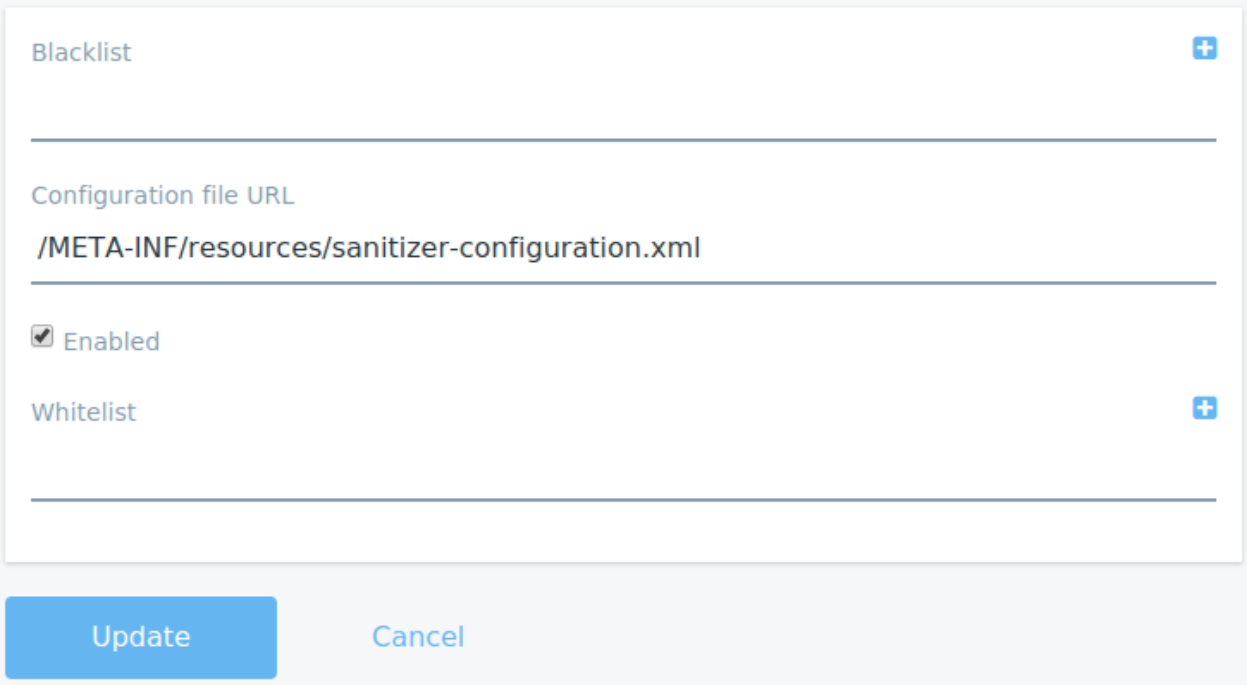


Figure 41.18: Liferay DXP's AntiSamy configuration options allow you to specify both a blacklist and a whitelist.

Using both a blacklist and a whitelist allows you to easily define subsets of entities that should be sanitized or not sanitized. By default, everything is sanitized except for `JournalArticle`. `JournalArticle` is the entity name for web content articles in Liferay DXP. But suppose you don't want to sanitize Liferay DXP entities except for message board posts. This might be reasonable if your portal only allows message boards posts to be created or updated by untrusted users.

To achieve this, you can

1. Configure the whitelist to *
2. Configure the blacklist to `com.liferay.message.boards.*`

Remember to use both the whitelist and the blacklist. Without the blacklist, you would need to configure the whitelist to include everything except `com.liferay.message.boards`: `com.liferay.journal.*`, `com.liferay.blogs.*`, etc.

41.21 Authenticating with Kerberos

You can use Kerberos to authenticate Microsoft Windows ™ accounts with Liferay DXP. This is done completely through configuration by using a combination of Liferay DXP's LDAP support and a web server that supports the Kerberos protocol.

Note that this configuration is preferred above NTLM because security vulnerabilities persist.

While it's beyond the scope of this article to explain how to set up Kerberos and Active Directory on a Windows ™ server, we can describe the minimum prerequisites for setting up Liferay authentication:

1. A Windows ™ server with Active Directory and DNS set up so the AD server and Liferay DXP can resolve each other on the network. In other words, they must be able to ping each other *by name*.

2. An administrative user in AD Liferay DXP can use to bind to AD.

3. A Kerberos keytab file exported via the ktpass command containing the cryptographic information the Liferay DXP server needs to bind to AD.

4. A web server in front of Liferay DXP that supports Kerberos, such as Apache, NGNIX, or IIS. The web server must also support injecting a header to be used as a token in the Liferay DXP configuration (see below).

5. Of course, you need a Liferay DXP installation that can also resolve by name the other servers. It should never run on the Active Directory server.

When you have all of these prerequisites in place, you're ready to configure Kerberos authentication.

How Kerberos Authentication Works

From the prerequisites, you may be able to guess that there are several moving parts to how SSO works with Kerberos.

First, a properly configured web browser sends a negotiate request using encrypted Windows user data. To configure this, the browser must recognize the site as a trusted site (explained below). The web server's Kerberos module uses the keytab file to bind over the Kerberos protocol to AD and verify the user information. If all is okay, the AD server confirms the connection with a valid response.

The web server you choose must support both the Kerberos protocol and the injection of a custom header into the request that Liferay DXP can later read. When the web server forwards the request to Liferay DXP, it reads the header to obtain the user data and authenticate the user.

Next, you'll learn how to get all of this working.

Configuring Kerberos Authentication

There are four components to configure: a user keytab from Active Directory, a web server in front of your application server, Liferay DXP, and your Windows™ clients.

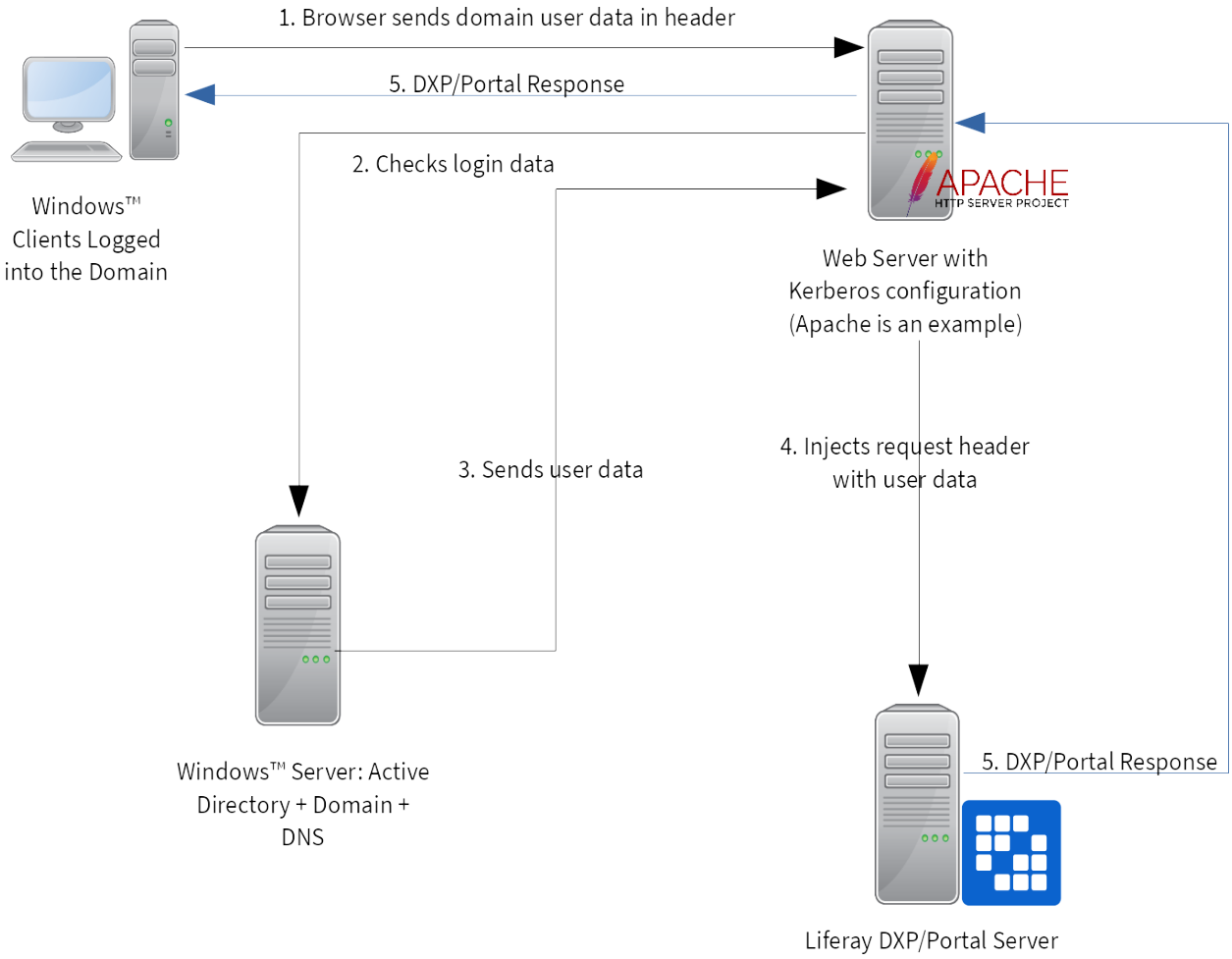


Figure 41.19: Kerberos authentication requires a web server in front of your Liferay DXP server.

Creating the User Keytab

1. Create a user so Liferay DXP can bind to Active Directory.

2. Generate a Kerberos keytab file using ktpass:

```
ktpass -princ HTTP/[web server host name]@[domain] -mapuser [user name]@[domain] -crypto ALL -ptype KRB5_NT_PRINCIPAL -
pass [password] -out c:\kerberos.keytab
```

   For example:

```
ktpass -princ HTTP/mywebserver.intdomain.local@INTDOMAIN.LOCAL -mapuser Marta@INTDOMAIN.LOCAL -crypto ALL -ptype KRB5_NT_PRINCIPAL -
pass password-for-Marta -out c:\kerberos.keytab
```

3. Ensure that the AD domain controller and the web server can see each other on the network via DNS configuration or hosts file.

Configuring Your Web Server

1. Configure Kerberos authentication. On Linux, this involves installing krb5 and configuring it to match your realm that's already configured for Active Directory. The example domain for the user configured in step two above would look like this:

```
[libdefaults]
    default_realm = INTDOMAIN.LOCAL

[domain_realm]
    mywebserver.intdomain.local = INTDOMAIN.LOCAL
    intdomain.local = INTDOMAIN.LOCAL
    .intdomain.local = INTDOMAIN.LOCAL

[realms]
INTDOMAIN.LOCAL = {
    admin_server = winserver.intdomain.local
    kdc = winserver.intdomain.local
}
```

2. Copy the keytab file you generated on your AD server to your web server.

3. Configure your web server, making sure you set the correct server name, Kerberos service name, Kerberos authentication realms, and the path to the keytab file. For example, if you're using the Apache HTTP server, the configuration might look like this:

```
LoadModule headers_module /usr/lib/apache2/modules/mod_headers.so
LoadModule rewrite_module /usr/lib/apache2/modules/mod_rewrite.so
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_http_module /usr/lib/apache2/modules/mod_proxy_http.so
LoadModule proxy_ajp_module /usr/lib/apache2/modules/mod_proxy_ajp.so
LoadModule auth_kerb_module /usr/lib/apache2/modules/mod_auth_kerb.so

<VirtualHost *:10080>
    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>
    ProxyRequests     Off
    ProxyPreserveHost On
    ProxyPass / ajp://localhost:8009/
    ProxyPassReverse / ajp://localhost:8009/
    ServerName mywebserver.intdomain.local
    <Location />
                Order allow,deny
                Allow from all
                AuthType Kerberos
                KrbServiceName HTTP/mywebserver.intdomain.local@INTDOMAIN.LOCAL
                AuthName "Domain login"
                KrbAuthRealms INTDOMAIN.LOCAL
                Krb5KeyTab /etc/apache2/kerberos.keytab
                require valid-user
                KrbMethodNegotiate  On
                KrbMethodK5Passwd   Off
                #KrbLocalUserMapping On

                # Below directives put logon name of authenticated user into http header X-User-Global-ID
                RequestHeader unset X-User-Global-ID
                RewriteEngine On
                RewriteCond   %{LA-U:REMOTE_USER} (.+)
                RewriteRule   /.* - [E=RU:%1,L,NS]
                RequestHeader set X-User-Global-ID %{RU}e
```

```
                # Remove domain suffix to get the simple logon name
                #RequestHeader edit X-User-Global-ID "@INTLAND.LOCAL$" ""

    </Location>
</VirtualHost>
Listen 10080
```

Connecting Liferay DXP to Active Directory over LDAP

1. Finally, configure Liferay DXP to access Active Directory via the LDAP protocol. Change authentication to be by Screen Name by selecting it in Configuration → Instance Settings → Authentication → General.

2. Connect Liferay DXP to AD over LDAP by going to Configuration → Instance Settings → Authentication → LDAP and adding an LDAP server. Provide the information appropriate to your installation:

   Base Provider URL: Your AD server on the proper port.

   Base DN: Your domain configuration. The example above might be `DC=INTDOMAIN.DC=LOCAL`.

   Principal/Credentials: Supply the credentials for the user exported to the keytab file.

   Authentication Search Filter: Supply the appropriate search filter to return user objects. For example, `(&(objectCategory=person)(sAMAccountName=*))`

   UUID: Supply what uniquely identifies a user, such as `sAMAccountName`.

   Screen Name: Supply the field that should be mapped to Liferay DXP's screen name field, such as `sAMAccountName`.

   Password: Supply the field that contains the user's password, such as `userPassword`.

3. Be sure to test the connection, save, and enable the configuration.

4. Finally, configure the token for single sign-on at Configuration → System Settings → Foundation → Token Based SSO. Make sure the User Token Name matches *exactly* the token you configured in your web server. Click the *Enabled* and *Import from LDAP* boxes and click *Save*.

Excellent! You've configured your servers. All that's left is to configure your clients.

Configuring your Clients

You must do two things: make your computer log into the domain and configure your Liferay DXP server as a trusted Internet site.

1. Join your computer to your domain. In keeping with the example above, you'd make your computer a member of the `INTDOMAIN.LOCAL` domain.

2. Log in as a user in that domain.

3. Internet Explorer, Edge, and Chrome use the Windows™ settings for trusted sites. If you use these browsers, go to Internet Options → Security → Local Intranet Sites and add your Liferay DXP server's URL. For example, add `http://mywebserver.intdomain.local:10080`.

4. Firefox can be configured by typing `about:config` in its address bar. Search for the below two preferences and add the Liferay DXP server's URL as the value for both:

   - `network.negotiate-auth.delegation-uris`

- `network.negotiate-auth.trusted-uris`

After configuring these things, test your configuration by accessing Liferay DXP through the web server's URL. Since you are already logged into your client machine, you should be automatically logged into Liferay DXP without a user/password prompt.

Congratulations on configuring Kerberos with Liferay DXP!

Chapter 42

Enabling Antivirus Scanning for Uploaded Files

You can automatically scan any file uploaded to Liferay for viruses. When you enable the antivirus scanner, it checks files on upload to Liferay applications, such as Documents and Media, Message Boards, and more. Virus-infected files are reported for users to reject.

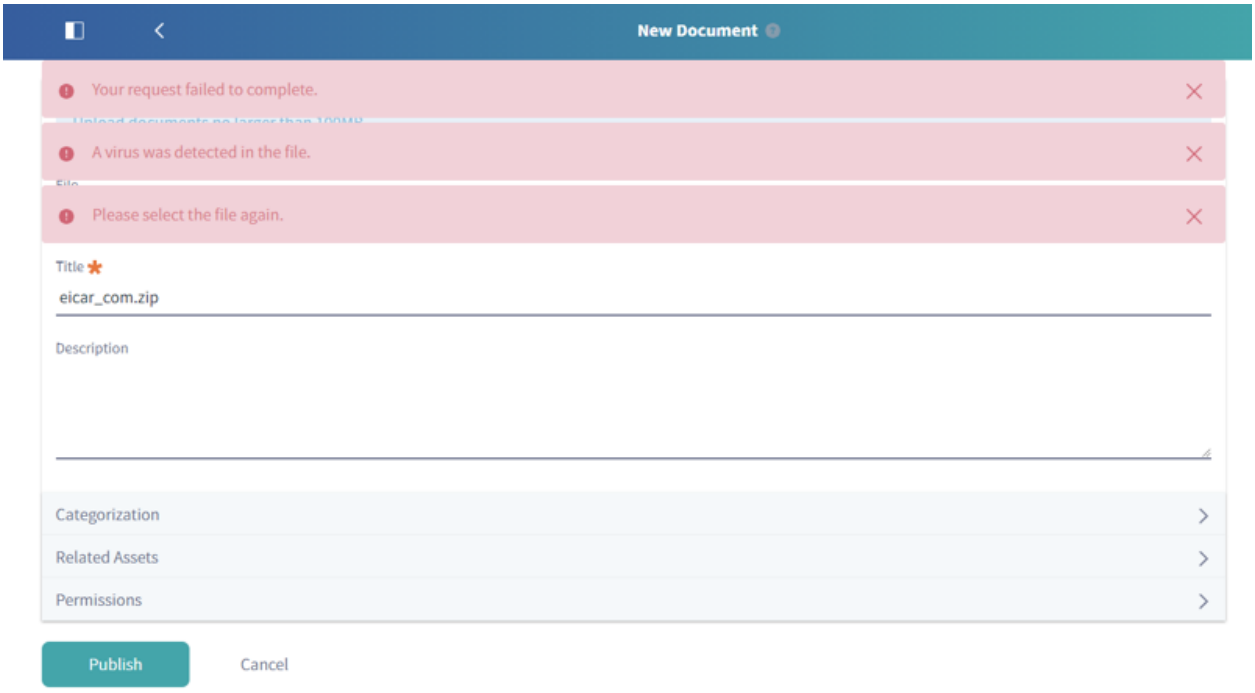


Figure 42.1: The scanner detects virus-infected files on upload to Documents and Media and other Liferay applications.

Liferay DXP 7.0 Fix Pack 96+ (and Service Pack 55+) integrates with the ClamAV Daemon (Clamd) running on a separate server.

Note: Prior to Liferay DXP 7.0 Fix Pack 96, the ClamAV antivirus scanner ran locally. Now Liferay DXP

delegates antivirus scanning to a separate server.

Here's how to enable the Clamd antivirus scanner:

1. On a separate server, configure and start Clamd.

   Important: Load your ClamAV database before starting Clamd.

2. Enable antivirus for your File Store (Document Library) by setting the following portal property or Docker Environment variable.

   Portal property:

   ```
   dl.store.antivirus.enabled=true
   ```

   Environment variable:

   ```
   -e LIFERAY_DL_PERIOD_STORE_PERIOD_ANTIVIRUS_PERIOD_ENABLED=true
   ```

3. Start your Liferay DXP server.

4. Go to *Control Panel* → *System Settings*, and select *Antivirus* in the *Security* category.

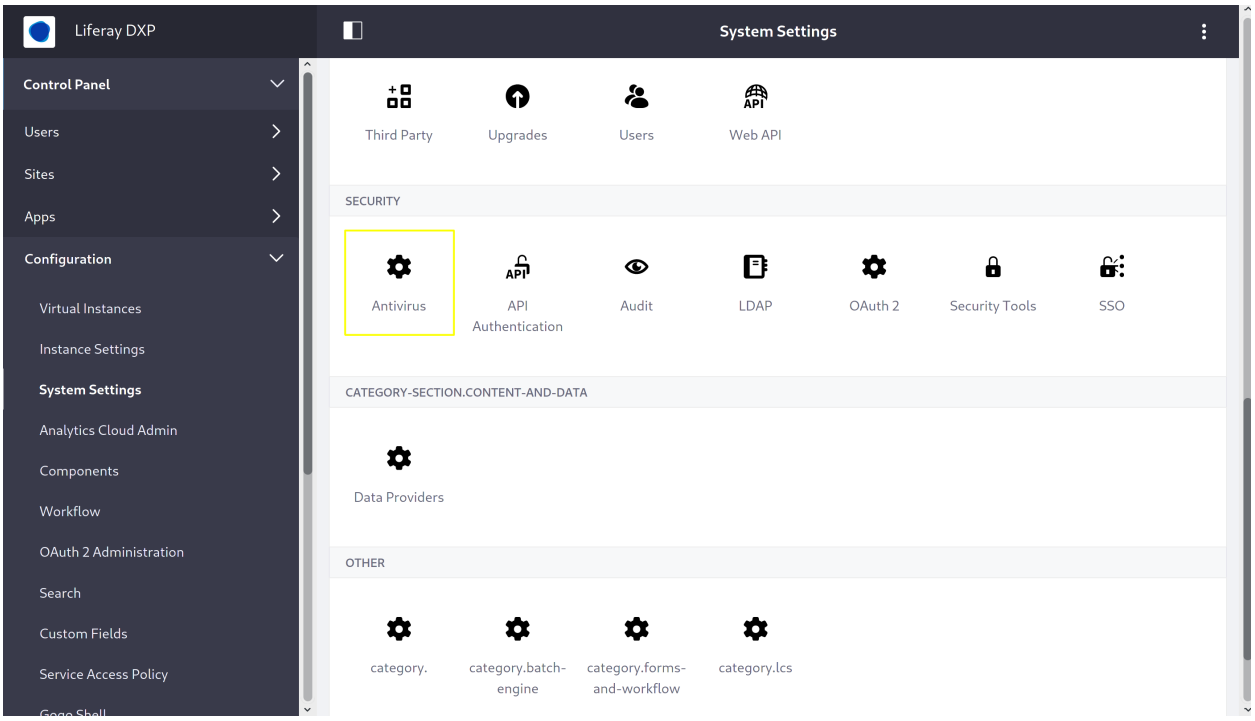


Figure 42.2: Antivirus is in the Security category in System Settings.

5. Select *Antivirus Clamd Scanner* in the menu.

6. Enter the Clamd server's hostname or IP address, port, and a connection timeout time (milliseconds).

7. Click *Save*.

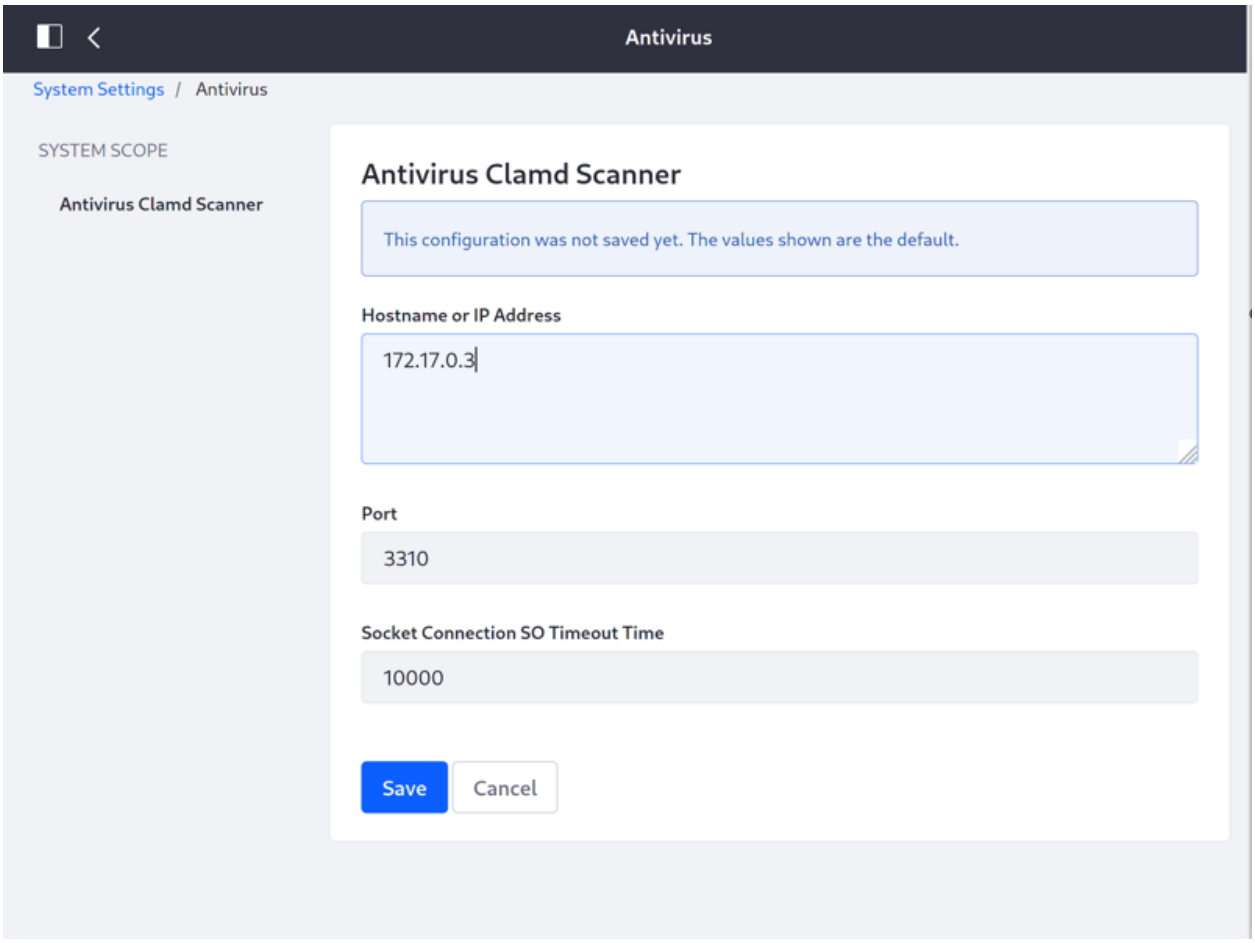


Figure 42.3: Antivirus Clamd Scanner configuration

Figure 42.4: Here's the virus detection message.

Now files are scanned on upload to Liferay applications. If a virus is detected in a file you're uploading, the scanner reports the infected file, you should decline saving the file.

Important: Never save a virus-infected file. Reject the file by canceling the current operation.

Chapter 43

Maintaining Liferay DXP

Once you have a Liferay DXP installation, there are some things you must do to keep it running smoothly. Backing up your installation in case of a hardware failure protects your data and helps you get your system back in working order quickly. And if you're a DXP customer, patching your system regularly brings the latest bug fixes to your running instance.

Upgrading 7.0 to a new GA version can require data upgrade. Until you perform all required data upgrades (if any), Liferay DXP startup fails with messages like these:

```
2018-11-05 17:22:35.025 INFO  [main][StartupHelper:72] There are no patches installed
You must first upgrade to 7.0
2018-11-05 17:22:35.098 ERROR [main][MainServlet:277] java.lang.RuntimeException: You must first upgrade to 7.0
```

Read on to learn about how to keep your system running well.

Chapter 44

Patching Liferay DXP

While we strive for perfection with every Liferay DXP release, the reality of the human condition dictates that releases may not be as perfect as originally intended. But we've planned for that. Included with every Liferay DXP bundle is a Patching Tool that handles installing two types of patches: fix packs and hotfixes.

Important: Make sure to back up your Liferay DXP installation and database regularly, especially before patching. The patching tool installs code changes and some of these make data changes (if necessary) automatically on startup.

Certain fix packs (service packs) can include data/schema micro changes—they're optional and revertible. Module upgrades are applied at server startup by default, or can be applied manually by disabling the `autoUpgrade` property. You can apply any upgrades using the upgrade tool before server startup.

Note: Patching a cluster requires additional considerations.

Chapter 45

Patching Basics

Liferay ships 7.0 fixes through three different channels:

- Fix packs
- Hotfixes
- Service Packs

45.1 Fix Packs

On a regular schedule, the latest fixes that patch the core are bundled together into fix packs that are provided to all of Liferay's customers. Fix packs include fixes for both the core and the application suites that ship with the product. Each fix pack contains all previous fix packs since the last service pack.

Security Fix Packs are a special fix pack type for deploying critical security fixes quickly without changing fix pack levels.

45.2 Hotfixes

A hotfix is provided to customers when they contact Liferay about an emergency situation, and Liferay's support team–working with the customer– determines that the problem is an issue with the product that needs to be fixed very quickly. Support fixes the bug and provides a hotfix to the customer immediately. This is a short-term fix Hotfixes can patch both the core and the application suites.

45.3 Service Packs

Service packs are built on the top of the original Liferay DXP release and repackaged with the latest fix pack, Patching Tool and modules. Since a service pack contains the latest fix pack, it contains all previous fix packs since the last service pack. If starting a new project, always start with the latest service pack.

Rather than use the service packs to keep existing systems updated, existing customers should

1. Keep their systems up-to-date with fix packs (according to their own deployment schedules).
2. Install the latest Marketplace updates frequently.

3. Update the Patching Tool when necessary.

This method updates the installation to the service pack levels, while allowing scheduled deployments and preventing full environment rebuilds.

45.4 How Patches are Tested

Liferay extensively tests all three types of fix packs to ensure high quality. For each issue fixed, fix packs go through both automated regression testing and manual testing. Hotfixes receive similar automated testing, and the fix for the reported issue is tested by the support engineer who fixed it.

Before releasing a service pack, test suites run on the packaged releases to ensure the quality of the packaging.

CHAPTER 46

CONFIGURING THE PATCHING TOOL

Liferay DXP patches can be installed by using the Patching Tool. In the prepackaged Liferay DXP installation (bundles), the Patching Tool is already configured. If you've installed Liferay DXP manually, however, you must also configure the Patching Tool manually. Read on to learn how to do that. If you're using a bundle, you can skip this section.

46.1 Patching Tool Basic configuration

There are two ways to configure the Patching Tool:

1. Automatically by executing the `auto-discovery` command
2. Manually by editing the configuration file (see the Advanced Configuration section)

Automatic configuration generates the configuration files by looking for Liferay DXP files in the file system. By default the Patching Tool looks for the Liferay DXP files in the parent folder. To start the process run

```
patching-tool auto-discovery
```

If Liferay DXP is not installed in the parent folder, you can specify its location:

```
patching-tool auto-discovery /opt/Liferay/tomcat-8.0.32
```

That's it! Now that you've installed the Patching Tool and run auto-discovery, you're ready to download and install patches. You can install patches manually or automatically.

If you specified the wrong folder or Liferay DXP is not installed in the parent folder, the Patching Tool won't be able to find the Liferay Home folder and shows an error:

```
The .liferay-home has not been detected in the given directory tree.

Configuration:
patching.mode=binary
war.path=../tomcat-8.0.32/webapps/ROOT/
global.lib.path=../tomcat-8.0.32/lib/ext/
liferay.home=**[please enter manually]**

The configuration hasn't been saved. Please save this to the default.properties file.
```

In this case you can either add the folder manually to the configuration or create the `.liferay-home` file and re-run the auto-discovery process.

When the patching tool is configured, you can run `patching-tool info` and receive information about the product version.

46.2 Patching Tool Advanced Configuration

By default, the Patching Tool's configuration file is in its installation folder and called `default.properties`.

A Patching Tool configuration file typically looks like this:

```
patching.mode=binary
war.path=../tomcat-8.0.32/webapps/ROOT/
global.lib.path=../tomcat-8.0.32/lib/ext/
liferay.home=../
```

The properties above (described fully below) define the location of Liferay Home, the patching mode (binary or source), the path to where WAR files are deployed in the app server, and the global library path. If auto-discovery found your Liferay Home folder, the location of Liferay DXP's OSGi-based module framework can be calculated from this. If, however, you customized the folder structure, you'll have to specify manually the following properties:

```
module.framework.core.path=path_to_modules_core_dir
module.framework.marketplace.path=path_to_modules_marketplace_dir
module.framework.modules.path=path_to_modules_modules_dir
module.framework.portal.path=path_to_modules_portal_dir
module.framework.static.path=path_to_modules_static_dir
```

For most installations, you don't have to do this, as the `osgi` folder is in its default location. If you've customized the location of the module framework, however, you'll have to specify the above locations. Since you moved them, you should know where they are.

Using Profiles with the Patching Tool

When you ran the auto-discovery task after installing the Patching Tool, it created a default profile that points to the application server it discovered. This is the easiest way to use the Patching Tool, and is great for smaller, single server installations. But many Liferay DXP installations are sized to serve millions of pages per day, and the Patching Tool has been designed for this as well. So if you're running a small, medium, or large cluster of Liferay DXP machines, you can use the Patching Tool to manage all of them using profiles.

The auto-discovery task creates a properties file called `default.properties`. This file contains the detected configuration for your application server. But you're not limited to only one server which the tool can detect. You can have it auto-discover other runtimes, or you can manually create new profiles yourself.

To have the Patching Tool auto-discover other runtimes, you must use a few more command line parameters:

```
./patching-tool.[sh|bat] [name of profile] auto-discovery [path/to/runtime]
```

This runs the same discovery process, but on a path you choose, and the profile information goes into a `[name of profile].properties` file.

Alternatively, you can manually create your profiles. Using a text editor, create a `[profile name].properties` file in the same folder as the Patching Tool script.

Below is a description of all the supported properties.

Configuration Properties

patching.mode: This can be `binary` (the default) or `source`, if you're patching a source tree. Liferay DXP patches contain both binary and source patches. If your development team is extending Liferay DXP, you'll want to provide the patches you install to your development team so they can patch their source tree.

patches.folder: Specify the location where you'll copy your patches. By default, this is `./patches`.

war.path: Specify the location of the Liferay DXP installation inside your application server. Alternatively, you can specify a .war file here, and you can patch a Liferay DXP .war for installation to your application server.

global.lib.path: Specify the location where `.jar` files on the global classpath are stored. If you're not sure, search for your `portal-kernel.jar` file; it's on the global classpath. This property is only valid if your `patching.mode` is `binary`.

liferay.home: Specify the location where by default the `data`, `osgi`, and `tools` folders reside.

source.path: Specify the location of your Liferay DXP source tree. This property is only valid if your `patching.mode` is `source`.

Note: To patch the 7.0 source code, please upgrade to Patching Tool 2.0.4+.

Service Pack detection is available behind a proxy server. Use the following settings to configure your proxy:

```
### Proxy settings

# HTTP Proxy

#proxy.http.host=192.168.211.39
#proxy.http.port=80
#proxy.http.user=user
#proxy.http.password=password

# HTTPS Proxy

proxy.https.host=192.168.211.39
proxy.https.port=808
proxy.https.user=user
proxy.https.password=password

# SOCKS Proxy

#proxy.socks.host=192.168.211.39
#proxy.socks.port=1080
#proxy.socks.user=user
#proxy.socks.password=password
```

You can have as many profiles as you want and use the same Patching Tool to patch all of them. This helps to keep all your installations in sync.

46.3 Installing patches on the 7.0 WAR

Because of your app server choice, you may not be able to patch the Liferay DXP instance that's installed, because the files aren't available on the file system. Instead, you must patch the Liferay DXP WAR file and then re-deploy it. This tutorial shows you how to do that.

Prerequisites

Download the necessary artifacts from the Customer Portal:

- Liferay DXP WAR (liferay-dxp-digital-enterprise-[version].war)
- OSGi dependencies (liferay-dxp-digital-enterprise-dependencies-[version].zip)
- Additional dependencies (liferay-dxp-digital-enterprise-osgi-[version].zip)
- Latest Patching Tool

How to Install a Fix Pack on the Liferay DXP WAR

1. Create a folder and unzip the dependency artifacts and the Patching Tool into it. Content of this folder should look like this:

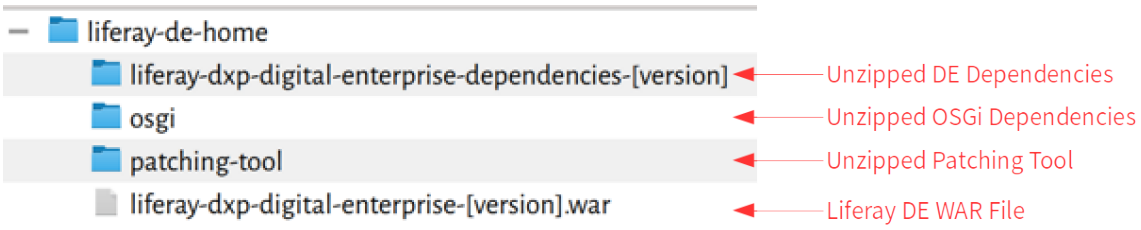


Figure 46.1: Use a simple folder structure for patching.

2. Create the default profile configuration file in the Patching Tool's home folder: `liferay-de-home/patching-tool/default.properties`. The contents should look like this:

```
patching.mode=binary
war.path=/liferay-de-home/liferay-dxp-digital-enterprise-[version].war
global.lib.path=/liferay-de-home/liferay-dxp-digital-enterprise-dependencies-[version]/
liferay.home=/liferay-de-home/
```

   If you have different OSGi folder structure, you can specify them as described in the documentation Patching Tool Advanced Configuration:

```
module.framework.core.path=d:/liferay-de-home/osgi/core
module.framework.marketplace.path=d:/liferay-de-home/osgi/marketplace
module.framework.modules.path=d:/liferay-de-home/osgi/modules
module.framework.portal.path=d:/liferay-de-home/osgi/portal
module.framework.static.path=d:/liferay-de-home/osgi/static
```

3. Download the patch (fix pack or hotfix) that you'd like to install and place it to the Patching Tool's patches folder without unzipping it.

4. Now the Patching Tool info command shows the following output:

```
/liferay-de-home/patching-tool>patching-tool.bat info

Loading product and patch information...
Product information:
  * installation type: binary
  * build number: 7010
  * service pack version:
```

```
    - available SP version: Not available
    - installable SP version: 1
  * patching-tool version: 2.0.5
  * time: 2016-12-05 13:29Z
  * plugins: no plugins detected

Currently installed patches: -

Available patches: de-8-7010

Detailed patch list:
  [ I] de-8-7010 :: Currently not installed; Will be installed.
```

5. Install the downloaded patch.

```
/liferay-de-home/patching-tool>patching-tool.sh install
One patch is ready to be installed. Applying de-8...
Cleaning up: [1%..10%..20%..30%..40%..50%..60%..70%..80%..90%..100%]
Installing patches: [1%..10%..20%..30%..40%..50%..60%..70%..80%..90%...100%]
The installation was successful. One patch is installed on the system.
```

Great! You have successfully patched the artifacts, and they are ready to be deployed on any supported Application Server.

Related Topics

Patching Tool Advanced Configuration
Deploying Liferay DXP
Installing Liferay DXP on WebLogic 12c R2
Installing Liferay DXP on WebSphere 8.5.5

Chapter 47

Using the Patching Tool

If you're using a Liferay DXP bundle, the Patching Tool is already installed. When an update is necessary to install a patch, the Patching Tool automatically asks for an update.

You follow the same procedure whether you're installing or upgrading the Patching Tool. Once you've obtained it from the Customer Portal, unzip it anywhere in the file system. Generally it's a good idea to keep it together with the Liferay DXP installation.

Upgrading is easy: override the previous Patching Tool with newest one by unzipping it on top of the old version.

47.1 Executables

The Patching Tool is a Java based application. The distribution contains shell/ .bat scripts to make it easier to use. On Unix systems you can run

```
./patching-tool.sh parameters
```

On Windows, run

```
patching-tool parameters
```

The latter method appears in the examples below. On Unix, replace the name of the executable before running the scripts.

47.2 Installing Patches

The first thing you must do when installing patches is to shut down your server. On Windows operating systems, files that are in use are locked by the OS, and can't be patched. On Unix-style systems, you can usually replace files that are running, but that still leaves the old ones loaded in memory. So your best bet is to shut down the application server that's running Liferay DXP before you install a patch.

Liferay distributes patches as `.zip` files, whether they are hotfixes or fix packs. When you receive one, either via a Help Center ticket (hotfix) or through downloading a fix pack from the Customer Portal, place it in the Patching Tool's `patches` folder (e.g., `[Liferay Home]/patching-tool/patches`) without unzipping it. Once you've done that, it's a simple matter to install it. First, execute

```
patching-tool info
```

This displays a list of patches you've already installed, along with a list of patches that *can* be installed from what's in the patches folder.

To install the available patches, use the following steps. First, issue the following command:

```
patching-tool install
```

To make sure the all changed OSGi bundles replace the existing ones, it is recommended to delete the osgi/state folder from the Liferay Home folder.

Important: The osgi/state folder should ONLY be deleted when working in a development environment or when applying a fix pack or hot fix.

Note: The osgi/state folder in the contains OSGi bundle state information. If an OSGi bundle's changes in a hot fix or fix pack are internal only and are, therefore, invisible to the OSGi framework, that OSGi bundle stays installed and its state information stays unchanged. Hot fixes, for example, may contain in-place changes that do not use the API—the framework cannot detect such changes. A fix pack's changes may also be transparent to the framework. For these reasons, deleting the osgi/state folder after applying fix packs and hot fixes is recommended.

If there are new database indexes created by the patch, the Patching Tool tells you to update them. To get the list, run this command:

```
patching-tool index-info
```

Since there's no database connection at patching time, the indexes must be created at portal startup. To have the indexes created automatically, add the following line to the portal-ext.properties file if the server has permissions to modify the indexes on the database:

```
database.indexes.update.on.startup=true
```

Otherwise, you must create the indexes manually. Check the output of the patching-tool index-info command for more details.

Once your patches have been installed, you can verify them by using the patching-tool info command, which now shows your patches in the list of installed patches.

Note: If there are any issues with the installed fixes, verify that there aren't any remaining files from the previous patch installation of a fix pack or hotfix within the application server cache.

During the installation, patching-backup-deps.zip and patching-backup.zip files are created and stored in the ROOT/WEB-INF folder. These files are necessary to restore the Liferay DXP's original state; removing them would disable further patching.

Note: When installing patches, Liferay DXP's web.xml is always overwritten by the one contained in the patch. If you've customized web.xml, you must re-implement your customizations after installing a patch.

The patching-backup.zip file is necessary for installing future patches, because the Patching Tool reverts the installed fix pack before installing a new one. To revert the installed fix pack, it examines the contents of the patching-backup.zip to determine the changes that it needs to revert.

Handling Hotfixes and Patches

As stated above, hotfixes are short term fixes provided as quickly as possible, and fix packs are larger bundles of hotfixes provided to all customers at regular intervals. If you already have a hotfix installed and the fix pack that contains that hotfix is released, the Patching Tool can manage this for you. Fix packs always supersede hotfixes, so when you install your fix pack, the hotfix that it contains is uninstalled, and the fix pack version is installed in its place.

Sometimes there can be a fix to a fix pack. This is also handled automatically. If a new version of a fix pack is released, you can use the Patching Tool to install it. The Patching Tool uninstalls the old fix pack and installs the new version in its place.

Including `support-info' in Help Center Tickets

To enable Liferay to reproduce subscriber issues, it is critical that the patch level in a given environment be made available to Liferay.

To generate the patch level for your environment, use the following command:

```
patching-tool support-info
```

A text file called `patching-tool-support-info-actual-timestamp.txt` is created in the patching-tool folder. Please upload this file to the Help Center ticket.

Fix Pack Dependencies

Some hotfixes require a fix pack to be installed first. If you attempt to install a hotfix that depends on a fix pack, the Patching Tool notifies you. Go to the customer portal and obtain the hotfix dependency. Once all the necessary patches are available in the `patches` folder, the Patching Tool installs them.

The Patching Tool can also remove patches.

47.3 Removing or Reverting Patches

Have you noticed that the Patching Tool only seems to have an `install` command? This is because patches are managed not by the command, but by what appears in the `patches` folder. You manage the patches you have installed by adding or removing patches from this folder. If you currently have a patch installed that you don't want, remove it from the `patches` folder. When you run the `patching-tool install` command, and the patch is removed.

If you want to remove all patches you've installed, use the `./patching-tool.sh revert` command. This removes all patches from your installation.

Prior to Fix Pack 13, the OSGi state folder could retain obsolete bundles in its cache. If you're running a version prior to Fix Pack 13, delete the *osgi/state* folder in Liferay Home.

47.4 Cleaning Up

After you've performed your patching procedure (whether you've installed or removed patches), it's important to clean up Liferay DXP's cache of deployed code. This ensures that when you start the server, you're using the revision you've just installed the patches for. This is really easy to do.

In the Liferay Home folder is a folder called `work`. Remove the contents of this folder to clear out the cached code. Now you're ready to start your server.

47.5 Comparing Patch Levels

If you're a developer, the Patching Tool can show you what changed between different versions. These commands show you information about the different patch levels:

`patching-tool diff`: Prints the differences between two patch levels. At least one stored patch level must be available. This command accepts options for filtering the output:

- `source`: Shows the source differences between the two patch levels.
- `files`: Shows a list of the modified files.
- `fixed-issues`: Shows a list of LPS/LPE issues from our issue tracking system.

For detailed usage information, run `patching-tool help diff`.

`patching-tool store`: Manages patching level information for the `diff` command. Your patches must contain source code to store the patch level and to prepare usable information for the `diff` command. Here are the `store` command options:

- `info`: Prints the list of patches which make up the stored patch level.
- `add`: Stores the patch level that can be found in the patches directory.
- `update`: Adds or updates patch level information.
- `rm`: Removes previously stored patch level information.

For detailed usage information, run `patching-tool help store`.

47.6 Showing collisions between patches and deployed plugins

Some patches update files you might have customized via a plugin. The `patching-tool list-collisions` command lists differences (collisions) between installed patch files and your plugin's version of them. Here's the command:

```
patching-tool list-collisions
```

It is an alias for the following diff command:

```
patching-tool diff collisions files _base
```

`_base` is the literal patch level name. Collisions are only listed for installed patches that contain source code files.

Note: As of Patching Tool 2.0.9, `patching-tool list-collisions` lists only JSP file collisions in fragment bundles.

47.7 Separating the Patches from the Liferay DXP Installation

As of Patching Tool 2.0.6, there's a feature that helps reduce the patched Liferay DXP bundle size. If the bundle has been patched, you can make it smaller by moving the restore files out of it.

Patched bundles are large because the restore files by default are stored inside the web application's WEB-INF folder. These files are required for patching the Liferay DXP instance again.

If these files were removed, subsequent patching processes would fail. Because of this, Liferay added an option to separate the patching files from the Liferay DXP bundle while still preserving restoring them safely when new patches arrive. To do this, you use this command:

```
patching-tool separate [separation_name]
```

This command produces a `liferay-patching-files-[separation-name].zip`file in the Patching Tool's `patches` folder. It contains the necessary files and metadata for patching, verification, and validation. Once you create this file, the patch files are removed from their default location and are now only available in this file. You can now move the file elsewhere to make the bundle's size smaller.

WARNING: If the product is separated from its patches in this way, you cannot run most of the Patching Tool commands until the patches are restored.

After the separation process only the following commands can be used:

- auto-discovery
- info
- setup

Any other command returns this:

```
This installation does not include data for patching. Please copy the
liferay-patching-files-[separation-name].zip file into the 'patches' directory
and run patching-tool setup.
```

This is how you restore the patch files to your system. Details below.

Restoring the Separated Patch Files

When you need to patch Liferay DXP again, you must restore the separated patch artifact. To do this, copy the `liferay-patching-files-[separation-name].zip` back to the Patching Tool's `patches` folder and run `patching-tool setup` command.

If the command finds the necessary patching artifact, it restores the patch files to the bundle. After that, the Patching Tool works like it did prior to separating the patches.

CHAPTER 48

KEEPING UP WITH FIX PACKS AND SERVICE PACKS

The *Announcements* section on Liferay's Help Center page lists all fix pack updates, security alerts, product releases, and system updates. The approximate frequency of fix pack and service pack releases is explained here. The *Receive Notifications* sidebar lets you subscribe to the latest updates on products, patches, and system improvements.

Click *Downloads* on the Liferay Digital Experience Platform page to access:

- Latest Release
- Fix Packs
- Service Packs Archive
- Security Advisories
- Patching Tool

Click *Support Information* to access the compatibility matrix, support FAQs, and more.

48.1 Securing Liferay DXP's Remote Services

Liferay DXP includes a script console which administrators can use to invoke Liferay DXP's API. See the Using Liferay DXP's Script Console article for more information. Liferay DXP's API can also be invoked remotely. How is this possible? Liferay DXP includes a utility called the *Service Builder* which is used to generate all of the low level code for performing CRUD operations for entities that are saved to Liferay DXP's database. This utility is further explained in the tutorial Service Builder. Service Builder generates web service interfaces that can be invoked remotely. Since Liferay DXP's APIs are generated by Service Builder, the method signatures and behavior for storing and retrieving Liferay DXP objects are consistent.

Because the actual method calls for retrieving data are the same regardless of how one gets access to those methods (i.e., locally or through web services), Liferay DXP provides a consistent interface for accessing portal data that few other products can match. The actual interfaces for the various services can be found by navigating to Liferay DXP's JSON web services page: http://localhost:8080/api/jsonws. Before these services can be invoked, administrators need to enable users to access these services remotely.

Liferay DXP includes four layers of web service security:

- IP permission layer
- Service access policy layer

- Authentication/verification layer (browser-only):
- User permission layer

Please see the Service Access Policies documentation for a description of these security layers.

In the default `portal.properties` file, there is a section called **Main Servlet**. This section defines the security settings for all of the remote services provided by Liferay DXP. Copy this section and paste it into your custom `portal-ext.properties` file. Then you can edit the default values to configure the security settings for the Axis Servlet, the Liferay Tunnel Servlet, the Spring Remoting Servlet, the JSON Tunnel Servlet, and the WebDAV servlet.

By default, a user connecting from the same machine Liferay DXP is running on can access remote services so long as that user has the permission to use those services in Liferay DXP's permissions system. Of course, you are not really "remote" unless you are accessing services from a different machine. Liferay DXP has multiple layers of security when it comes to accessing its services remotely. Without explicit rights granted for each layer, attempting to invoke a Liferay DXP API function remotely fails and a remote exception is thrown.

The first layer of web service security that a user needs to get through in order to call a method from the service layer is servlet security. This is the IP permission layer listed above. The *Main Servlet* section of the `portal-ext.properties` file is used to enable or disable access to Liferay DXP's remote services. In this section of the properties file, there are properties for each of Liferay DXP's remote services.

You can set each service individually with the security settings that you require. For example, you may have a batch job which runs on another machine in your network. This job looks in a particular shared folder on your network and uploads documents to your site's Documents and Media portlet on a regular basis, using Liferay DXP's web services. To enable this batch job to get through the first layer of security, you would modify the `portal-ext.properties` file and put the IP address of the machine on which the batch job is running in the list for that particular service. For example, if the batch job uses the Axis web services to upload the documents, you would enter the IP address of the machine on which the batch job is running to the `axis.servlet.hosts.allowed` property. A typical entry might look like this:

```
axis.servlet.hosts.allowed=192.168.100.100, 127.0.0.1, SERVER_IP
```

If the machine on which the batch job is running has the IP address `192.168.100.100`, this configuration will allow that machine to connect to Liferay DXP's web services and pass in user credentials to be used to upload the documents.

The second layer of web service security is the service access policy layer. When attempting to invoke a Liferay DXP API function remotely, the function must be whitelisted by one of the service access policies that apply to the request.

The third layer of security is the authentication/verification layer. This layer is only applied to a Liferay DXP API function invocation request if the request is coming from a browser. In this case, the request must include a valid authentication token.

The fourth and final layer of security is Liferay DXP's permissions system that protects every Liferay DXP entity. The user account that accesses the services remotely must have the proper permissions to operate on the objects it attempts to access. Otherwise, a remote exception is thrown. Portal administrators need to use Liferay DXP's permissions system to grant access to these resources to the administrative user account that attempts to operate on them remotely. For example, suppose that a Documents and Media folder called *Documents* has been set up in a site. A role has been created called *Document Uploaders* which has the rights to add documents to this folder. Your batch job will be accessing Liferay DXP's web services in order to upload documents into this folder. In order for this to work, you have to call the web service using a user account to which the *Document Uploaders* role has been assigned (or that has individual rights to add documents to the folder). Otherwise, you will be prevented from using the web service.

You can invoke a Liferay DXP web service in a variety of ways. For example, you can invoke a web service by navigating to a specific URL in your browser, by using the cURL tool, or via JavaScript. Here's an example URL for a Liferay DXP web service invocation using a browser:

```
http://localhost:8080/api/jsonws/user/get-user-by-email-address/company-id/20154/email-address/test%40liferay.com?p_auth=[value]
```

This Liferay DXP web service invocation returns the user with the email address `test@example.com` in the company (portal instance) with the ID `20154`. Note that for the web service invocation to succeed, you must supply a correct value for the `p_auth` URL parameter. You can find such a value by signing in to Liferay DXP, navigating to http://localhost:8080/api/jsonws and clicking on any function that appears in the list. Use the value for the `p_auth` token that appears under the Execute heading. For more examples of Liferay DXP web service invocations, please see the JSON Web Services Invocation Examples documentation.

Some methods of invoking Liferay DXP web services (e.g., using cURL) allow you to supply a username and password. Note that *Password Policies* (covered in the User Management documentation) can be used in combination with this feature. If you are enforcing password policies on your users (requiring passwords to take a certain form, requiring users to change their passwords on a periodic basis, etc.), any administrative user account which accesses Liferay DXP's web services in a batch job can have its password expire too.

To prevent this from happening, you can add a new password policy which does not enforce the password expiration and add your administrative user ID to it. Then your batch job can run as many times as you need it to and the administrative user account's password will never expire.

In summary, accessing Liferay DXP remotely requires the successful passing of four security checks:

1. The IP address must be pre-configured in the server's `portal-ext.properties` file.
2. At least one service access policy which applies to the request must have the API function being invoked in a whitelist.
3. If a browser is making the web service invocation request, a valid authentication token (`p_auth` URL parameter) must be provided.
4. The user ID being used must have permission to access the resources it attempts to access.

Accessing Liferay DXP's JSON Web Services

To see which Liferay DXP service methods are registered and available for use via JSON web services, open your browser to the following address:

```
http://localhost:8080/api/jsonws
```

The page lists the portal's registered and exposed service methods. Get each method's details by clicking the method name. You can view the full signature of each method, all its arguments, the exceptions that can be thrown, and its Javadoc! Using a simple form from within your browser, you can even invoke the service method for testing purposes.

To list registered services on a plugin (e.g. a custom portlet), don't forget to use its context path in your URL:

```
http://localhost:8080/[plugin-context]/api/jsonws
```

This lists the JSON Web Service API for the plugin.

Accessing Liferay DXP's WSDL

After configuring the security settings properly, your first step in obtaining access to Liferay DXP's remote SOAP web services is to access the WSDL. If you are on a browser on the same machine Liferay DXP is running on, you can do this by accessing the following URL:

```
http://localhost:[port number]/tunnel-web/axis
```

If, for example, you are running on Tomcat on port 8080, you would specify this URL:

```
http://localhost:8080/tunnel-web/axis
```

If you are accessing a web service that was created as part of a portlet plugin, the URL is similar, but uses the context of your application rather than the tunnel-web servlet. You can get a list of your Service Builder-generated WSDL documents by using the URL pattern below:

```
http://localhost:8080/your-portlet/axis
```

If you are on a different machine from the Liferay DXP server, you will need to pass in your user credentials on the URL to access the WSDL:

```
http://[user ID]:[password]@[server name]:[port number]/tunnel-web/axis
```

In any case, once you successfully browse to this URL, you will see the list of web services. You can access the WSDL for each service by clicking on the *WSDL* link next to the name of the service. There are many services; one for each of the services available from the Liferay DXP API.

Once you click on one of the *WSDL* links, the Web Service Definition Language document will be displayed. This document can be used to generate client code in any language that supports it. You can either save the document to your local machine and then generate the client code that way, or use your tool to trigger Liferay DXP to generate the document dynamically by using one of the URLs above. For information about developing applications that leverage Liferay DXP's remote services, please see the Invoking JSON Web Services documentation.

Chapter 49

Backing up a Liferay DXP Installation

Once you have an installation of Liferay DXP running, you should implement a comprehensive backup plan. In case some kind of catastrophic hardware failure occurs, you'll be thankful to have backups and procedures for restoring Liferay DXP from one of them. Liferay DXP isn't very different from other Java web application that might be running on your application server. Nevertheless, there are some specific components you should include in your backup plan.

The recommended backup plan includes backing up these things:

- Source code
- Liferay DXP's file System
- Liferay DXP's database

49.1 Backing up Source Code

If you have extended Liferay DXP or have written any plugins, they should be stored in a source code repository such as Git, Subversion, or CVS, unless you're Linus Torvalds, and then tarballs are okay too (that's a joke). You should back up your source code repository on a regular basis to preserve your ongoing work. This probably goes without saying in your organization since nobody wants to lose source code that's taken months to produce. Thus you should include source code in your Liferay DXP backup plan.

Next, let's examine the Liferay DXP installation items you should back up.

49.2 Backing up Liferay's File System

The Liferay Home folder stores Liferay DXP's properties configuration files, such as `portal-setup-wizard.properties` and `portal-ext.properties`. You should absolutely back them up. In fact, it's best to back up your entire application server and Liferay Home folder contents.

Liferay DXP stores configuration files, search indexes, and cache information in Liferay Home's `/data` folder. If you're using the File System store or the Advanced File System store, the documents and media repository is also stored here by default. It's always important to back up your `/data` folder.

Important: If you've configured the document library to store files to a location other than a `[Liferay Home]/data` subfolder, back up that location.

The files that comprise Liferay DXP's OSGi runtime are stored in Liferay Home's `/osgi` folder. It contains all of the app and module JAR files deployed to Liferay DXP. The `/osgi` folder also contains other required JAR files, configuration files, and log files. It's also important to back up your `/osgi` folder.

Liferay Home's `/logs` folder contains Liferay DXP's log files. If a problem occurs on Liferay DXP, the Liferay DXP log files often provide valuable information for determining what went wrong. The `/data`, `/osgi`, and `/logs` folders are all contained in the Liferay Home folder. Thus, if you're backing up both your application server folder and your Liferay Home folder, you're in good shape.

That covers the Liferay DXP file system locations you should back up. Next, let's discuss how to back up Liferay DXP's database.

49.3 Backing up Liferay DXP's Database

Liferay DXP's database is the central repository for all of the portal's information. It's the most important component to back up. You can back up the database live (if your database allows this) or by exporting (dumping) the database into a file and then backing up the exported file. For example, MySQL ships with a `mysqldump` utility which lets you export the entire database and data into a large SQL file. This file can then be backed up. On restoring the database you can import this file into the database to recreate the database state to that of the time you exported the database.

If you're storing Liferay DXP's Documents and Media Library files to a Jackrabbit JSR-170 repository database, you should back it up. If you've placed your search index into a database (not recommended; see the Liferay DXP Clustering article for information on using Cluster Link or Solr), you should back up that database too.

If you wish to avoid re-indexing your content after restoring your database, back up your search indexes. This is easiest to do if you have a separate Elastic or Solr environment on which your index is stored. If you're in a clustered configuration and you're replicating indexes, you'll need to back up each index replica.

Restoring your application server, your Liferay Home folder, the locations of any file system-based media repositories, and your database from a backup system should give you a functioning portal. Restoring search indexes should avoid the need to re-index when you bring your site back up after a catastrophic failure. Good, consistent backup procedures are key to recovering successfully from a hardware failure.

CHAPTER 50

UPGRADING TO 7.0

Upgrading to Liferay DXP consists of two main steps: upgrading your installation and then upgrading the database. Liferay DXP can be upgraded using a straightforward process. To upgrade to the latest release directly, you must be coming from Liferay Portal 6.0.12 or higher.

Note: The Liferay Upgrade Planner provides an automated way to adapt your installation's data and legacy plugins.

If you're on Liferay Portal 6.0.11 or below, you should upgrade to Liferay Portal 6.2 before approaching an upgrade to the 7.0 platform. Please see the Upgrading to Liferay Portal 6.2 article for information on upgrading to Liferay Portal 6.2 first.

Before you do anything, however, you should follow the pre-upgrade process of cleaning and normalizing your data. After that, you can prepare your system for the upgrade, and then finally run the upgrade process.

The tutorials that follow lead you through those steps.

50.1 Pre upgrade - Speed up the process

The most complex step in upgrading Liferay DXP is running the process that upgrades the database from the old version to the new version. It takes a long time to restructure the data to the new format.

You can shorten this process by performing a few steps before you upgrade your production environment. Here's a summary of the steps:

1. Copy your most recent complete backup from production to a non-production environment in which you can analyze your database and test upgrading, as explained in the remaining steps.
2. Examine your database.
3. Use Liferay's API to delete unused content.
4. Run the upgrade process on your non-production environment.
5. Check the upgrade log for the processes that took the most time.
6. Remove unused content from the upgrade processes that took the longest. If you see a potential issue, analyze it and contact the community if you need help. If you have an enterprise subscription, feel free to open a support ticket and have Liferay verify your analysis.
7. Repeat steps 4, 5, and 6 as needed.
8. Remove unused content from production.

The sections that follow explain the more in-depth steps listed above.

Examine Your Database (Step 2)

Here are the most important things to examine in your non-production environment's copy of the production database:

- Records per table
- Size per table (optional)

The greater these values are the longer upgrade processes take.

The database engines show this information in different ways. Sometimes the output from importing backup data into your non-production database shows each table's size and number of rows (records).

For example, output from a typical database import looks like this:

```
Processing object type SCHEMA\_EXPORT/TABLE/TABLE\_DATA

imported "LIFERAY"."JOURNALARTICLE" 13.33 GB 126687 rows

imported "LIFERAY"."RESOURCEPERMISSION" 160.9 MB 1907698 rows

imported "LIFERAY"."PORTLETPREFERENCES" 78.13 MB 432285 rows

imported "LIFERAY"."LAYOUT" 52.05 MB 124507 rows

imported "LIFERAY"."ASSETENTRY" 29.11 MB 198809 rows

imported "LIFERAY"."MBMESSAGE" 24.80 MB 126185 rows

imported "LIFERAY"."PORTALPREFERENCES" 4.091 MB 62202 rows

imported "LIFERAY"."USER\_" 17.32 MB 62214 rows

imported "LIFERAY"."USERS\_ROLES" 15.63 MB 1117225 rows

imported "LIFERAY"."LAYOUTSET" 11.50 MB 124442 rows

imported "LIFERAY"."MBTHREAD" 11.99 MB 126184 rows

imported "LIFERAY"."COUNTER" 6.161 MB 123699 rows

imported "LIFERAY"."CONTACT\_" 5.233 MB 62214 rows

imported "LIFERAY"."GROUP\_" 3.772 MB 62221 rows

imported "LIFERAY"."ASSETTAGPROPERTY" 1.460 MB 21417 rows

imported "LIFERAY"."MBDISCUSSION" 3.138 MB 126179 rows

imported "LIFERAY"."JOURNALARTICLEIMAGE" 1.015 MB 22523 rows

imported "LIFERAY"."USERS\_GROUPS" 897.1 KB 62218 rows

imported "LIFERAY"."IMAGE" 480.9 KB 13492 rows

imported "LIFERAY"."JOURNALSTRUCTURE" 135.9 KB 12 rows

imported "LIFERAY"."JOURNALTEMPLATE" 230.6 KB 44 rows

imported "LIFERAY"."DLFILEENTRY" 195.3 KB 1028 rows

imported "LIFERAY"."DLFILEVERSION" 196.8 KB 1350 rows
```

```
imported "LIFERAY"."JOURNALARTICLERESOURCE" 272.6 KB 4665 rows

imported "LIFERAY"."JOURNALCONTENTSEARCH" 107.1 KB 2038 rows

imported "LIFERAY"."ASSETENTRIES\_ASSETTAGS" 90.89 KB 6118 rows

imported "LIFERAY"."COUNTRY" 15.33 KB 227 rows

imported "LIFERAY"."DLFILERANK" 12.83 KB 135 rows

imported "LIFERAY"."QUARTZ\_JOB\_DETAILS" 19.70 KB 2 rows

imported "LIFERAY"."QUARTZ\_TRIGGERS" 11.64 KB 1 rows

imported "LIFERAY"."TICKET" 147.9 KB 1666 rows

imported "LIFERAY"."ASSETTAG" 74.31 KB 908 rows

imported "LIFERAY"."CLASSNAME\_" 13.36 KB 148 rows

imported "LIFERAY"."COMPANY" 9.210 KB 1 rows

imported "LIFERAY"."RATINGSSTATS" 71.67 KB 2566 rows

imported "LIFERAY"."ACCOUNT\_" 11.35 KB 1 rows

imported "LIFERAY"."ANNOUNCEMENTSDELIVERY" 13.85 KB 192 rows

imported "LIFERAY"."ASSETCATEGORY" 11.17 KB 2 rows

imported "LIFERAY"."ASSETENTRIES\_ASSETCATEGORIES" 5.539 KB 1 rows

imported "LIFERAY"."ASSETTAGSTATS" 20.82 KB 666 rows

imported "LIFERAY"."ASSETVOCABULARY" 11.03 KB 6 rows

imported "LIFERAY"."DLFOLDER" 35.81 KB 195 rows

imported "LIFERAY"."LAYOUTPROTOTYPE" 7.359 KB 1 rows

imported "LIFERAY"."QUARTZ\_CRON\_TRIGGERS" 6.367 KB 1 rows

imported "LIFERAY"."RESOURCEACTION" 42.84 KB 806 rows

imported "LIFERAY"."SERVICECOMPONENT" 25.28 KB 1 rows

imported "LIFERAY"."SOCIALACTIVITY" 45.42 KB 734 rows

imported "LIFERAY"."WEBDAVPROPS" 7.820 KB 2 rows

imported "LIFERAY"."WIKIPAGE" 14.68 KB 1 rows

imported "LIFERAY"."EXPANDOTABLE" 6.656 KB 9 rows

imported "LIFERAY"."IGFOLDER" 10.17 KB 8 rows

imported "LIFERAY"."IGIMAGE" 18.61 KB 61 rows

imported "LIFERAY"."JOURNALFEED" 24.22 KB 38 rows

imported "LIFERAY"."LISTTYPE" 9.554 KB 63 rows

imported "LIFERAY"."MBCATEGORY" 10.75 KB 1 rows

imported "LIFERAY"."MBMAILINGLIST" 15.96 KB 1 rows

imported "LIFERAY"."MBMESSAGEFLAG" 7.304 KB 1 rows
```

```
imported "LIFERAY"."PASSWORDPOLICY" 19.06 KB 1 rows

imported "LIFERAY"."PORTLET" 11.15 KB 136 rows

imported "LIFERAY"."QUARTZ\_LOCKS" 5.117 KB 5 rows

imported "LIFERAY"."QUARTZ\_SCHEDULER\_STATE" 6 KB 1 rows

imported "LIFERAY"."REGION" 15.27 KB 316 rows

imported "LIFERAY"."RELEASE\_" 8.281 KB 1 rows

imported "LIFERAY"."ROLE\_" 17.17 KB 115 rows

imported "LIFERAY"."SHARD" 6.382 KB 1 rows

imported "LIFERAY"."SUBSCRIPTION" 8.648 KB 2 rows

imported "LIFERAY"."USERGROUPROLE" 5.992 KB 2 rows

imported "LIFERAY"."USERS\_ORGS" 5.531 KB 2 rows

imported "LIFERAY"."VIRTUALHOST" 6.398 KB 1 rows

imported "LIFERAY"."WIKINODE" 9.453 KB 2 rows

imported "LIFERAY"."WIKIPAGERESOURCE" 6.382 KB 1 rows
```

Several items stand out in the example database import:

- The *JOURNALARTICLE* table makes up 98% of the database size.
- There are many *RESOURCEPERMISSION* records.
- There are many *PORTLETPREFERENCES* records.

The table records reflect Liferay DXP objects. Using the API to delete objects that you no longer need not only deletes each object's data record but also deletes related unneeded objects (and their data records). For example, deleting an unneeded `Group` object also deletes related unneeded layouts, journal articles, and more. This reduces the number of records your upgrade needs to process, making your upgrade faster.

Use Liferay's API to remove unused objects (Step 3)

Your database analysis revealed tables that were large or contained lots of records. It's recommended to find unneeded objects that can be deleted by examining objects associated with these tables. Also there are some common areas (listed below) to look for unneeded objects.

Important: You should only use Liferay's API–Core API and app APIs– to delete objects because the API accounts for relationships between Liferay DXP objects. You can invoke the API through the Control Panel's script console or a portlet you create.

Never run SQL directly on your database to remove records. Your SQL might miss object relationships, resulting in orphaned objects and performance problems.

Here are some common areas to find unneeded objects:

- **Sites**: Remove sites you don't need. When you remove a site, remove its related data:
    - Layouts
    - Portlet preferences

  - File entries (document library objects)
  - Asset Entries
  - Tags
  - Vocabularies and categories
  - Expando fields and their values
  - `ResourcePermission` objects
  - (and everything else)

- **Instances**: Unused instances are rare, but since they are the highest object in the hierarchy, removing their objects can optimize upgrades considerably:

  - Sites (and all their related content)
  - Users
  - Roles
  - Organizations
  - Global `ResourcePermission` objects

- **Intermediate web content versions:** Liferay DXP generates a new web content version after any modification (including translations). Consider removing versions you don't need. Removing a Journal Article, for example, also removes related objects such as image files (`JournalArticleImage`) that are part of the content. Removing unneeded image files frees space in your database and file system.

- **Document versions**: As with Journal Articles, if you don't need intermediate document versions, delete them. This saves space both in the database and on the file system, space that no longer needs to be upgraded.

- **Layouts:** Layouts are site pages, and they affect upgrade performance because they relate to other entities such as portlet preferences, permissions, assets, ratings, and more. Remove unneeded layouts.

- **Roles**: Remove any roles you don't need. Deleting them also deletes related `ResourceBlockPermission` and `ResourcePermission` objects.

- **Users:** If you have users that aren't active anymore, remove them.

- **Vocabularies**: Remove any unused vocabularies. Note that removing a vocabulary also removes its categories.

- **Orphaned data**: Check for unused objects that are not connected to anything. Here are some examples:

  - `DLFileEntries` with no file system data.
  - `ResourcePermission` objects associated to a role, layout, user, portlet instances, etc. that no longer exists.
  - `PortletPreference` objects associated with a portlet or layout that no no longer exists. This is common in environments with many embedded portlets. These portlet instances have a different lifecycle, and aren't deleted when the portlet is removed from a template.

After you've removed unneeded objects, test your changes.

Execute the upgrade process (Step 4)

It's time to upgrade your non-production environment and note what upgrade processes take the longest. Each Liferay DXP upgrade process logs how long it takes. An upgrade log now looks like this:

```
21:42:45,422 INFO \[main\]\[UpgradeProcess:83\] Upgrading com.liferay.portal.upgrade.v7\_0\_0.UpgradeRatings

21:42:45,423 INFO \[main\]\[LoggingTimer:70\] Starting com.liferay.portal.upgrade.v7\_0\_0.UpgradeRatings\#upgradeRatingsEntry

21:42:47,154 INFO \[main\]\[LoggingTimer:38\] Completed com.liferay.portal.upgrade.v7\_0\_0.UpgradeRatings\#upgradeRatingsEntry in 1731 ms

21:42:47,154 INFO \[main\]\[LoggingTimer:70\] Starting com.liferay.portal.upgrade.v7\_0\_0.UpgradeRatings\#upgradeRatingsStats

21:44:10,069 INFO \[main\]\[LoggingTimer:38\] Completed com.liferay.portal.upgrade.v7\_0\_0.UpgradeRatings\#upgradeRatingsStats in 82915 ms

21:44:10,070 INFO \[main\]\[UpgradeProcess:98\] Completed upgrade process com.liferay.portal.upgrade.v7\_0\_0.UpgradeRatings in 84648ms
```

The duration times (in milliseconds) facilitate finding the most time consuming processes. Consider searching for unneeded objects associated these longer upgrade processes. Once again, make sure to delete them using Liferay's API and test your changes.

Note: Learning how upgrade processes are created can help you understand their data better.

Remove Unused Objects from Production (Step 8)

Now that you have removed unused objects from your non-production environment and tested your changes, you can use Liferay's API to remove the same objects from your production environment. By removing the objects from production and testing your changes before upgrading, you can more easily troubleshoot any issues, knowing that they're not related to upgrade processes. Another benefit of doing this even while you work through the upgrade is that your production environment will perform better and be easier to maintain.

Conclusion

By removing unused objects from Liferay DXP, you can both reduce upgrade time and improve your server's performance on the new version.

Taking the time to optimize your installation before upgrading can save time and keep your installation running smoothly.

50.2 Preparing an Upgrade to 7.0

Now that you've completed the pre-upgrade process of cleaning and normalizing your data, you're ready to prepare your environment for upgrading to 7.0. Here's a summary of the preparation steps:

Step 1: Upgrade your Marketplace apps
Step 2: Publish all changes from the staged site to the live site
Step 3: Remove duplicate web content structure field names
Step 4: Synchronize a complete Liferay DXP backup
Step 5: Update your portal properties
Step 6: Configure your Documents and Media file store
Step 7: Install 7.0 and the latest fix pack
Step 8: Disable indexing during the upgrade process
This tutorial describes these steps in detail.

Step 1: Upgrade Your Marketplace Apps

Upgrade each Marketplace app (Kaleo, Calendar, Notifications, etc.) that you're using to its latest version for your installation. Before proceeding with the upgrade, troubleshoot any issues regarding these apps.

Step 2: Publish All Changes from the Staged Site to the Live Site

If you have local/remote staging enabled and have content or data saved on the staged site, you must publish it to the live site. If you skip this step, you must run a full publish (or manually publish changes) after the upgrade, since the system won't know what content changed since the last publishing date.

Step 3: Remove Duplicate Web Content Structure Field Names

If you've used Web Content Management extensively, you might have structures whose field names aren't unique. You must find and remove duplicate field names before upgrading. If you upgraded to Liferay Portal 6.2 previously and skipped doing this, you'll encounter this error:

```
19:29:35,298 ERROR [main][VerifyProcessTrackerOSGiCommands:221] com.liferay.portal.verify.VerifyException: com.liferay.dynamic.data.mapping.validator.DDMFor
com.liferay.portal.verify.VerifyException: com.liferay.dynamic.data.mapping.validator.DDMFormValidationException$MustNotDuplicateFieldName: The field name p
```

If this is the case, roll back to your previous backup of Liferay 6.2 and find and remove duplicate field names.

Step 4: Synchronize a Complete Backup Liferay DXP

Back up your Liferay DXP database, installation, and Document Library store.

Step 5: Update Your Portal Properties

It is likely that you have overridden portal properties to customize your installation to your requirements. If so, you must update the properties files (e.g., `portal-setup-wizard.properties` and `portal-ext.properties` files) to be compatible with 7.0. As you do this, you should account for property changes in all versions of Liferay DXP since your current version up to and including 7.0.

If you're coming from a version prior to Liferay Portal 6.2, start with these property-related updates:

- If you're on Liferay Portal 6.1, adapt your properties to the new defaults that Liferay Portal 6.2 introduced.
- If you're on Liferay 6.0.12, migrate the Image Gallery.
- If you have a sharded environment, configure your upgrade for sharding.

When a new version of Liferay DXP is released, there are often changes to default settings, and this release is no different. If you rely on the defaults from your old version, you'll want to review the changes and decide whether you want to keep the defaults from your old version or accept the defaults of the new.

Here's a list of the 6.2 properties that have changed in 7.0:

```
users.image.check.token=false
organizations.types=regular-organization,location
organizations.rootable[regular-organization]=true
organizations.children.types[regular-organization]=regular-organization,location
organizations.country.enabled[regular-organization]=false
organizations.country.required[regular-organization]=false
organizations.rootable[location]=false
```

```
#organizations.children.types[location]=
organizations.country.enabled[location]=true
organizations.country.required[location]=true
layout.set.prototype.propagate.logo=true
editor.wysiwyg.portal-web.docroot.html.taglib.ui.discussion.jsp=simple
web.server.servlet.check.image.gallery=true
blogs.trackback.enabled=true
discussion.comments.format=bbcode
discussion.max.comments=0
dl.file.entry.thumbnail.max.height=128
dl.file.entry.thumbnail.max.width=128
```

Properties in features that have been modularized have changed and must now be deployed separately in OSGi configuration files. The 7.0 portal properties reference docs provide property details and examples.

Step 6: Configuring Your Documents and Media File Store

It's time to migrate and update your document store configuration to 7.0. Here's what's changed for document stores:

1. Store implementation class package names changed from `com.liferay.portlet.documentlibrary.store.*` in Liferay Portal 6.2 to `com.liferay.portal.store.*` in Liferay DXP 7.0+. Make sure your `portal-ext.properties` sets the `dl.store.impl` property one of these ways:

   ```
   dl.store.impl=com.liferay.portal.store.file.system.FileSystemStore
   dl.store.impl=com.liferay.portal.store.db.DBStore
   dl.store.impl=com.liferay.portal.store.file.system.AdvancedFileSystemStore
   dl.store.impl=com.liferay.portal.store.s3.S3Store
   ```

2. CMIS Store and JCR Store were deprecated as of Liferay DXP Digital Enterprise 7.0 Fix Pack 14 (SP3) and Liferay Portal CE 7.0 GA4. The Document Repository Configuration documentation describes other store options. Migrate your document data to one of the other store options before upgrading from your current Liferay version.

3. If you're using Advanced File System Store or Simple File System Store, make the store accessible to your new installation. For example, copy the store files to the location `[Liferay Home]/data/document_library` in your new installation.

4. Since Liferay DXP 7.0, document store specific configuration (e.g., configurations specific to Simple File Store, Advanced File Store, S3, etc.) is done in the Control Panel at *Configuration → System Settings → File Storage* or done using OSGi configuration files (`.config` files).

   Note, general document store configuration (e.g., `dl.store.impl=[File Store Impl Class]`) continues to be done using `portal-ext.properties`.

   Here are steps, for example, to configure an existing Advanced File System Store in your new installation:

   1. Create a `.config` file named after the store implementation class (i.e., the class assigned to your `dl.store.impl` property):

      ```
      com.liferay.portal.store.file.system.configuration.AdvancedFileSystemStoreConfiguration.config
      ```

   2. Set the following property in the `.config` file and replace `{document_library_path}` with your Advanced File Store path.

```
rootDir="{document_library_path}"
```

3. Copy the `.config` file to your new installation's `[Liferay Home]/osgi/configs` folder.

Important: If you're using Advanced File System Store, you must configure it with a `.config` file in the new installation before upgrading the database.

The Document Repository Configuration documentation provides more information.

Step 7: Install 7.0

Next, install Liferay DXP on your application server or use Liferay DXP bundled with your application server of choice.

Then if you're upgrading Liferay DXP, install the latest fix pack.

Important: Once you have installed 7.0, **DON'T START IT!** In Liferay Portal 6.2 and earlier, once you prepared your system for an upgrade, the upgrade process ran when you started the new version for the first time. Now to streamline server startup, Liferay DXP ships with an upgrade tool (described in the next article) that you must use to upgrade your database.

Note: Liferay DXP throws the following exception if you attempt to start 7.0 before running the upgrade tool:

```
[MainServlet:237] java.lang.RuntimeException: You must first upgrade to Liferay DXP 7000
```

Copy your custom portal properties files (e.g., `portal-ext.properties`) that you updated in previous steps and your Documents and Media store into your new installation.

Step 8: Disable Indexing During the Upgrade Process

Before starting the upgrade process in your new installation, you must disable indexing to prevent upgrade process performance issues that arise when the indexer attempts to reindex content.

To disable indexing, create a file called `com.liferay.portal.search.configuration.IndexStatusManagerConfiguration.conf` in your `[Liferay Home]/osgi/configs` folder and add the following content:

```
indexReadOnly="true"
```

After you complete the upgrade (described in the next article), re-enable indexing by removing the `.config` file or setting `indexReadOnly="false"`.

Ready to upgrade? The next article shows you how.

50.3 Running the Upgrade Process

Now you're ready to run the upgrade process. It updates the database schema for the core and your installed modules. Verification processes test the upgrade. Configured verifications for the core and modules run afterwards, but can be run manually too.

Here are the ways to run upgrade processes:

- **Upgrade everything in one shot**: Use the upgrade tool to upgrade the core and all the modules.

- **Upgrade the core and the modules separately**: Use the upgrade tool (recommended) or Gogo shell to upgrade the core. Then use Gogo shell to upgrade each module.

If you are upgrading from Liferay Portal 6.2 or earlier, it's recommended to use the upgrade tool to upgrade everything. It's the easiest, most comprehensive way to upgrade from those versions. For version 7.0 onward, however, Liferay DXP's modular framework lets you upgrade modules–even the core–individually. Focusing first on upgrading the core and your most important modules might be better for you. The point is, 7.0 upgrade process is flexible.

Running the Upgrade Tool

The upgrade tool provides the easiest way to upgrade the core and installed modules. You can configure the upgrade from files or inside the tool's command line interface. The upgrade tool lets you upgrade everything–the core and all the modules–together or separately.

7.0 bundles include the upgrade tool. If you installed @product-ver@ manually, you can download the upgrade tool separately.

- *Liferay DXP 7.0*: Go to the *Downloads* page, select product *DXP 7.0* and file type *Product*, and select *Download* for *Liferay DXP DB Upgrade Client*.

- *Liferay Portal CE 7.0*: Go to SourceForge, select *7.0 GA[version]*, and click `liferay-ce-portal-tools-[version].zip`.

Before running the upgrade tool, learn the tool's usage and how to configure the core upgrade and non-core module upgrades.

- Upgrade Tool Usage
- Configuring Non-Core Module Upgrades
- Configuring the Core Upgrade

Start with the tool's usage.

Upgrade Tool Usage

The `db_upgrade.sh` script (`db_upgrade.bat` on Windows) invokes the upgrade tool. It resides in the `[Liferay Home]/tools/portal-tools-db-upgrade-client` folder.

This command prints the upgrade tool usage:

```
db_upgrade.sh --help
```

To upgrade only the core, add a file called `com.liferay.portal.upgrade.internal.configuration.ReleaseManagerConfigurat` to the `[Liferay Home]/osgi/configs` folder with the following content:

```
autoUpgrade="false"
```

This configuration prevents automatic module upgrade, but causes the upgrade tool to open a Gogo shell for upgrading modules after finishing the core upgrade.

Here are the tool's default Java parameters:

```
-Dfile.encoding=UTF-8 -Duser.country=US -Duser.language=en -Duser.timezone=GMT -Xmx2048m
```

The `-j` option lets you override the JVM parameters. For example, these options set the JVM memory to 10GB, which is a good starting point for this process type:

```
db_upgrade.sh -j "-Dfile.encoding=UTF-8 -Duser.country=US -Duser.language=en -Duser.timezone=GMT -Xmx10240m"
```

The `-l` option lets you specify the tool's log file name:

```
db_upgrade.sh -l "output.log"
```

Here are all the upgrade tool command line options:
–help or **-h**: Prints the tool's help message.
–jvm-opts or **-j** + **[arg]**: Sets any JVM options for the upgrade process.
–log-file or **-l** + **[arg]**: Specifies the tool's log file name—the default name is `upgrade.log`.
–shell or **-s**: Automatically connects you to the Gogo shell after finishing the upgrade process.

Note: Only execute the upgrade process on a server with ideal memory, CPU, and database connection configuration. If executing an upgrade remotely using `ssh`, make sure to guard against interruptions:

- If you're executing the upgrade using `ssh`, ignore hangups (connection loss) by using `nohup` or something similar.
- On the machine you're connecting from, disable settings that shutdown or sleep that machine.

Since DB Upgrade Tool 2.0.1, the upgrade process continues on the server even if you lose connection to it. If you lose connection, reconnect and monitor upgrade status via the log (default log file is `upgrade.log`). If you're using an earlier version of 7.0 and upgrade execution is interrupted, check your log file for where execution stopped.

- If execution stopped during an upgrade process for any module upgrade process, restart the upgrade tool to continue the upgrade from that point. You can also use Gogo shell to check module upgrade status and continue upgrading modules.
- If execution stopped during an upgrade process for Core 7.0 or lower, you must restore the data from a backup and start the upgrade again.

Warning: To prevent the tool's expanded command from growing too large for Windows, execute the upgrade tool script from the `[Liferay Home]/tools/portal-tools-db-upgrade-client` folder.

Before starting the upgrade, decide how to execute non-core module upgrades.

Configuring Non-Core Module Upgrades

You can configure the upgrade tool to upgrade all installed modules automatically or to open a Gogo shell (after core upgrade completes) for you to execute module upgrades manually.

If the upgrade tool's `autoUpgrade` property is set to `true` (the default setting), upgrade processes for all installed modules are run too.

If you set `autoUpgrade="false"` in a file called `com.liferay.portal.upgrade.internal.configuration.ReleaseManagerConfigu` and copy the file into the `[Liferay Home]/osgi/configs` folder, the upgrade tool opens Gogo shell after the core upgrade. In the Gogo shell, you can administer module upgrades.

Now that you've decided how to do non-core module upgrades, examine the core upgrade configuration options.

Configuring the Core Upgrade

The core upgrade requires configuration. You can configure it at runtime via the command line interface or pre-configure it in these files in `[Liferay Home]/tools/portal-tools-db-upgrade-client/`:

- `app-server.properties`: Specifies the server's location and libraries.
- `portal-upgrade-database.properties`: Configures the database connection.
- `portal-upgrade-ext.properties`: Sets the rest of the portal properties that the upgrade requires. You might want to copy your current portal properties (except your database properties) into this file. Before copying your current properties, make sure you've updated the portal properties for 7.0.

Each file's properties are described next.

Configuring app-server.properties

Specify the following information to configure the app server on which 7.0 is installed:

`dir`: the absolute path of the application server directory. *(required)*

`extra.lib.dirs`: a comma delimited list of extra directories containing any binaries or resources to add to the class path. Use relative paths to `dir`. *(required)*

`global.lib.dir`: the application server's global library directory. Use a path relative to `dir`. *(required)*

`portal.dir`: the directory where portal is installed in your app server. Use a path relative to `dir`. *(required)*

`server.detector.server.id`: ID of a supported application server. (*required*) Here are the IDs:

- `jboss`
- `jonas`
- `resin`
- `tomcat`
- `weblogic`
- `websphere`
- `wildfly`

Relative paths must use Unix style format (forward slashes) and start with a `/`. For example, the following properties are for Windows:

```
dir=D:\liferay-dxp\tomcat-8.0.32
extra.lib.dirs=/bin
global.lib.dir=/lib
portal.dir=/webapps/ROOT
server.detector.server.id=tomcat
```

As another example, the following properties are for Linux:

```
dir=/home/user/liferay
extra.lib.dirs=/liferay-portal-master/tomcat-8.0.32/bin
global.lib.dir=/liferay-portal-master/tomcat-8.0.32/lib
portal.dir=/liferay-portal-master/tomcat-8.0.32/webapps/ROOT
server.detector.server.id=tomcat
```

Configuring portal-upgrade-database.properties

Specify the following information to configure the database you're upgrading. Note that these properties correspond exactly to the JDBC portal properties you'd use in a `portal-ext.properties` file.

`jdbc.default.driverClassName` *(required)*
`jdbc.default.url` *(required)*
`jdbc.default.username` *(required)*
`jdbc.default.password` *(required)*

Configuring portal-upgrade-ext.properties

Specify the following information to configure the upgrade itself:

`liferay.home`: the Liferay home folder *(required)*

`locales`: Liferay upgrades data for all of the specified locales. If you added locales to the property in your `portal-ext.properties`, specify the updated property here.

`dl.store.impl`: the implementation for persisting documents to the document library store. This property's default value is `com.liferay.portal.store.file.system.FileSystemStore`. If you updated the property in your `portal-ext.properties` to use a different implementation, specify the updated property here.

`hibernate.jdbc.batch_size`: the JDBC batch size used to improve performance; set to *250* by default *(optional)*

Example Upgrade Configuration

Here's an example interaction with the upgrade tool's command line interface:

```
Please enter your application server (tomcat):
tomcat
Please enter your application server directory (../../tomcat-8.0.32):

Please enter your extra library directories (../../tomcat-8.0.32/bin):

Please enter your global library directory (../../tomcat-8.0.32/lib):

Please enter your portal directory (../../tomcat-8.0.32/webapps/ROOT):

[ db2 mariadb mysql oracle postgresql sqlserver sybase ]
Please enter your database (mysql):
mariadb
Please enter your database host (localhost):

(etc.)
```

The command line interface creates the configuration files based on your input. If you want to set all of this up ahead of time, however, you'll want to put this information into configuration files.

Here are example upgrade configuration files that you can customize and copy into `[Liferay Home]/tools/portal-tools-db-upgrade-client/`:

- `app-server.properties`:

  ```
  dir=../../tomcat-8.0.32
  global.lib.dir=/lib
  portal.dir=/webapps/ROOT
  server.detector.server.id=tomcat
  extra.lib.dirs=/bin
  ```

- `portal-upgrade-database.properties`:

```
jdbc.default.url=jdbc:mysql://lportal62?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDateParsin
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.username=root
jdbc.default.password=
```

- `portal-upgrade-ext.properties`:

```
liferay.home=/home/user/servers/liferay7
module.framework.base.dir=/home/user/servers/liferay7/osgi
dl.store.impl=com.liferay.portal.store.file.system.FileSystemStore
```

It's time to start the core upgrade.

Running and Managing the Core Upgrade

Start the upgrade tool, as explained in the upgrade tool usage. Here are the core upgrade stages:

1. Show the upgrade patch level
2. Execute the core upgrade processes
3. Execute the core verifiers

Monitor the upgrade via the upgrade tool log file (default file is `upgrade.log`). If a core upgrade process fails, analyze the failure and resolve it.

If you configured the upgrade tool to upgrade non-core modules, the tool opens a Gogo shell and starts upgrading them. The Gogo shell lets you upgrade modules, check module upgrade status, verify upgrades, and restart module upgrades. Read on to learn how to use Gogo shell commands to complete Liferay DXP upgrades.

Gogo shell commands for module upgrades

Liferay DXP's Gogo shell commands let you upgrade modules, check module status, or execute verify processes.

Note: Configuring the core upgrade is required before using Gogo shell commands to upgrade the core.

If you ran the upgrade tool and it opened Gogo shell, you're already connected. Otherwise, you can connect to Gogo shell via telnet:

```
telnet localhost 11311
```

Here are the commands:

Command	Description
`exit` or `quit`	Exits the Gogo shell
`upgrade:help`	Displays upgrade commands
`upgrade:check`	Lists upgrades pending execution because they failed in the past or the module hasn't reached its final version
`upgrade:execute [module_name]`	Executes upgrades for that module
`upgrade:executeAll`	Executes all pending module upgrade processes

Command	Description
`upgrade:list`	Lists all registered upgrades
`upgrade:list [module_name]`	Lists the module's required upgrade steps
`upgrade:list | grep Registered`	Lists registered upgrades and their versions
`verify:help`	Displays verify commands
`verify:check [module_name]`	Lists the latest execution result for the module's verify process
`verify:checkAll`	Lists the latest execution results for all verify processes
`verify:execute [module_name]`	Executes the module's verifier
`verify:executeAll`	Executes all verifiers
`verify:list`	Lists all registered verifiers

There are many useful Liferay commands and standard commands available in Gogo shell. The following sections describe Liferay upgrade commands.

Listing module upgrade processes

Before upgrading modules, you should find which have unresolved dependencies, which are resolved and available to upgrade, and examine the module upgrade processes.

Executing `upgrade:list` in the Gogo shell lists the modules whose upgrade dependencies are satisfied. These modules can be upgraded.

If a module is active but not listed, its dependencies need to be upgraded. The Gogo shell command `scr:info [upgrade_step_class_qualified_name]` shows the upgrade step class's unsatisfied dependencies. Here's an example `scr:info` command:

```
scr:info com.liferay.journal.upgrade.JournalServiceUpgrade
```

Invoking `upgrade:list [module_name]` lists the module's upgrade processes, in no particular order. For example, executing `upgrade:list com.liferay.bookmarks.service` (for the Bookmarks Service module), lists this:

```
Registered upgrade processes for com.liferay.bookmarks.service 1.0.0
        {fromSchemaVersionString=0.0.0, toSchemaVersionString=1.0.0, upgradeStep=com.liferay.portal.spring.extender.internal.context.ModuleApplicationContex
        {fromSchemaVersionString=0.0.1, toSchemaVersionString=1.0.0-step-3, upgradeStep=com.liferay.bookmarks.upgrade.v1_0_0.UpgradePortletId@5f41b7ee}
        {fromSchemaVersionString=1.0.0-step-1, toSchemaVersionString=1.0.0, upgradeStep=com.liferay.bookmarks.upgrade.v1_0_0.UpgradePortletSettings@53929b1d
        {fromSchemaVersionString=1.0.0-step-2, toSchemaVersionString=1.0.0-step-1, upgradeStep=com.liferay.bookmarks.upgrade.v1_0_0.UpgradeLastPublishDate@3
        {fromSchemaVersionString=1.0.0-step-3, toSchemaVersionString=1.0.0-step-2, upgradeStep=com.liferay.bookmarks.upgrade.v1_0_0.UpgradeClassNames@6964cb
```

An application's upgrade step class names typically reveal their intention. For example, the example's `com.liferay.bookmarks.upgrade.v1_0_0.UpgradePortletId` upgrade step class updates the app's portlet ID. The other example upgrade step classes update class names, the `LastPublishDate`, and `PortletSettings`. The example's step from `0.0.0` to `1.0.0` upgrades the module from an empty database.

To examine a module's upgrade process better, you can sort the listed upgrade steps mentally or in a text editor. Here's the upgrade step order for a Bookmarks Service module to be upgraded from Liferay Portal 6.2 (the module's database exists) to schema version `1.0.0`:

- `0.0.1` to `1.0.0-step-3`
- `0.0.1-step-3` to `1.0.0-step-2`
- `0.0.1-step-2` to `1.0.0-step-1`
- `0.0.1-step-1` to `1.0.0`

The overall module upgrade process starts at version `0.0.1` and finishes at version `1.0.0`. The first step starts on the initial version (`0.0.1`) and finishes on the target version's highest step (`step-3`). The last step starts on the target version's lowest step (`step-1`) and finishes on the target version (`1.0.0`).

Once you understand the module's upgrade process, you can execute it with confidence.

Executing module upgrades

Executing `upgrade:execute [module_name]` upgrades the module. You might run into upgrade errors that you must resolve. Executing the command again starts the upgrade from the last successful step.

You can check upgrade status by executing `upgrade:list [module_name]`. For example, entering `upgrade:list com.liferay.iframe.web` outputs this:

```
Registered upgrade processes for com.liferay.iframe.web 0.0.1
   {fromSchemaVersionString=0.0.1, toSchemaVersionString=1.0.0, upgradeStep=com.liferay.iframe.web.upgrade.IFrameWebUpgrade$1@1537752d}
```

The first line lists the module's name and current version. The example module's current version is `0.0.1`. The `toSchemaVersionString` value is the target version.

Executing `upgrade:list [module_name]` on the module after successfully upgrading it shows the module's name followed by the version you targeted.

For example, if you successfully upgraded `com.liferay.iframe.web` to version `1.0.0`, executing `upgrade:list com.liferay.iframe.web` shows the module's version is `1.0.0`:

```
Registered upgrade processes for com.liferay.iframe.web 1.0.0
   {fromSchemaVersionString=0.0.1, toSchemaVersionString=1.0.0, upgradeStep=com.liferay.iframe.web.upgrade.IFrameWebUpgrade$1@1537752d}
```

For module upgrades that don't complete, you can check their status and resolve their issues.

Checking upgrade status

It's good to know things still need upgrading and why. You might have forgotten to upgrade a module or its upgrade failed. In any case, it's important to know where your upgrade stands.

The command `upgrade:check` lists modules that have impending upgrades.

For example, if module `com.liferay.dynamic.data.mapping.service` failed in a step labeled `1.0.0-step-2`. Executing `upgrade:check` shows this:

```
Would upgrade com.liferay.dynamic.data.mapping.service from 1.0.0-step-2 to
1.0.0 and its dependent modules
```

Modules often depend on other modules to complete upgrading. Executing `scr:info [upgrade_step_class_qualified_name` shows the upgrade step class's dependencies. You must upgrade dependency modules to successfully upgrade dependent modules.

To resolve and activate a module, its upgrade must complete. The Apache Felix Dependency Manager Gogo shell command `dm wtf` reveals unresolved dependencies. If your module requires a certain data schema version (e.g., its `bnd.bnd` specifies `Liferay-Require-SchemaVersion: 1.0.2`) but the module hasn't completed upgrade to that version, `dm wtf` shows that the schema version is not registered.

```
1 missing dependencies found.
-------------------------------------
The following service(s) are missing:
 * com.liferay.portal.kernel.model.Release (&(release.bundle.symbolic.name=com.liferay.journal.service)(release.schema.version=1.0.2)) is not found in the s
```

The `dm wtf` command can also help detect errors in portlet definitions and custom portlet `schemaVersion` fields.

Browsing the Liferay DXP database `Release_` table can help you determine a module's upgrade status too. The core's `servletContextName` field value is `portal`. If the core's `schemaVersion` field matches your new Liferay DXP version (e.g., `7.0.1` for Liferay Portal CE GA2) and the `verified` field is `1` (true), the core upgrade completed successfully.

Each module has one `Release_` table record, and the value for its `schemaVersion` field must be `1.0.0` or greater (`1.0.0` is the initial version for 7.0 modules, except for those that were previously traditional plugins intended for Liferay Portal version 6.2 or earlier).

Executing verify processes

Verify processes make sure the upgrade executed successfully. Verify processes in the core are automatically executed after upgrading Liferay DXP. You can also execute them by configuring the `verify.*` portal properties and restarting your server.

Also, some modules have verify processes. To check for available verify processes, enter the Gogo shell command `verify:list`. To run a verify process, enter `verify:execute [verify_qualified_name]`.

Post-Upgrade Tasks

After upgrading and running verify processes, make sure to re-enable search indexing by either removing the `com.liferay.portal.search.configuration.IndexStatusManagerConfiguration.config` file from your `[Liferay Home]/osgi/configs` folder or setting this property in it:

```
indexReadOnly="false"
```

Then you should reindex Liferay DXP's search indexes. Don't just do this blindly, however. By default, Liferay DXP ships with an embedded configuration for Elasticsearch. This configuration works great for demo purposes, but is not supported in production. Make sure to install and configure a standalone Elasticsearch instance to run in production.

Once you've configured search and reindexed your search index, your upgraded system is ready for action! Congratulations!

50.4 Upgrading a Sharded environment

Liferay removed its own physical partitioning implementation (also known as sharding) in favor of the capabilities provided natively by database vendors. Upgrading a sharded installation to 7.0 requires migrating it to as many non-sharded Liferay DXP installations (servers) as you have shards. These steps guide you through configuring the new Liferay DXP servers to use your formerly sharded data and upgrading the data to 7.0.

Note: Liferay continues to support its logical partitioning capabilities (also known as virtual instances) for the foreseeable future.

For any further assistance with sharding contact your Liferay account manager or Liferay Support.

Here are the upgrade steps:

1. If you're on Liferay Portal 6.1 or lower, upgrade to Liferay Portal 6.2.
2. Remove unneeded objects from your portal.
3. Prepare for upgrading to 7.0.
4. Run the upgrade process for the default shard. As part of the configuration, copy all of the shard JDBC connection properties from your `portal-ext.properties` to your `portal-upgrade-database.properties`. For example, JDBC connections for a default shard and two non-default shards might look like this:

```
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.url=jdbc:mysql://localhost/lportal?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDa
jdbc.default.username=
jdbc.default.password=

jdbc.one.driverClassName=com.mysql.jdbc.Driver
jdbc.one.url=jdbc:mysql://localhost/lportal_one?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDa
jdbc.one.username=
jdbc.one.password=

jdbc.two.driverClassName=com.mysql.jdbc.Driver
jdbc.two.url=jdbc:mysql://localhost/lportal_two?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDa
jdbc.two.username=
jdbc.two.password=
```

5. Prepare a 7.0 server for each non-default shard.
6. Run the upgrade process for each non-default shard. The JDBC *default* connection properties in each server's `portal-upgrade-database.properties` must specify the associated shard. Here's how:

    - Add the original JDBC properties for the respective non-default shard database. For example, shard `one`'s original properties might start with `jdbc.one`:

    ```
    jdbc.one.driverClassName=com.mysql.jdbc.Driver
    jdbc.one.url=jdbc:mysql://localhost/lportal_one?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&us
    jdbc.one.username=
    jdbc.one.password=
    ```

    - Rename the properties to start with `jdbc.default`. For example,

    ```
    jdbc.default.driverClassName=com.mysql.jdbc.Driver
    jdbc.default.url=jdbc:mysql://localhost/lportal_one?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=tru
    jdbc.default.username=
    jdbc.default.password=
    ```

7. In each server's `portal-ext.properties`, use the JDBC *default* properties you specified in the `portal-upgrade-database.properties` (see the previous step).
8. Remove the non-default shard JDBC properties from the default shard server's `portal-ext.properties` file, leaving only the default shard database `jdbc.default` properties. For example,

    Old JDBC properties:

```
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.url=jdbc:mysql://localhost/lportal?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDa
jdbc.default.username=
jdbc.default.password=

jdbc.one.driverClassName=com.mysql.jdbc.Driver
jdbc.one.url=jdbc:mysql://localhost/lportal_one?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDa
```

```
jdbc.one.username=
jdbc.one.password=

jdbc.two.driverClassName=com.mysql.jdbc.Driver
jdbc.two.url=jdbc:mysql://localhost/lportal_two?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDa
jdbc.two.username=
jdbc.two.password=
```

New JDBC properties:

```
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.url=jdbc:mysql://localhost/lportal?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDa
jdbc.default.username=
jdbc.default.password=
```

Congratulations! You have migrated off of a sharded environment to virtual instances on separate Liferay DXP servers. You have also upgraded to 7.0. Your virtual instances are ready for action.

50.5 Upgrading Social Office

Liferay Social Office, Liferay's social collaboration product for the enterprise, was an add-on product for Liferay Portal versions prior to 7.0. Social Office is no longer available because @product-ver@ contains its features. When upgrading from a previous version of Liferay Portal that contains Social Office, the standard upgrade procedure handles most things for you. You must, however, perform a few additional steps to ensure that Social Office's features work as intended in 7.0. This article takes you through these steps.

Note: Before upgrading your Social Office installation, you must first upgrade your Liferay Portal installation to 7.0. The steps for this are the same as those for upgrading any Liferay Portal installation to 7.0; click here to see them. Once you've upgraded to 7.0, return here to upgrade your Social Office installation.

First, you'll learn how Social Office's unique components map to 7.0 features.

Social Office Components

Social Office contained the following components:

- A unique theme and site template
- Customized Liferay Portal applications
- Applications unique to Social Office
- User experience (UX) enhancements

The following sections describe the components and how they work in 7.0.

Themes and Templates

Social Office's unique look and feel was defined by its theme and site template. These don't exist in 7.0. In @product-ver@, however, you can build themes and templates to fit your requirements. The ability to customize the look, feel, and page layout of 7.0 is one of its most powerful capabilities.

This table shows how Social Office's theme and template components map to 7.0.

Component | Social Office 3.x | 7.0 | Social Office Theme | Uses Social Office theme | Uses default 7.0 theme, or a custom theme | Site Template | Uses Social Office site template | The Social Office site template

is upgraded to 7.0. A custom site template can also be used. | Page Templates | Doesn't use page templates | Page templates can be used |

Liferay Apps

Social Office improved many out-of-the-box apps in previous versions of Liferay Portal. Many of those improvements are now in 7.0.

This table shows how the apps modified by Social Office in previous versions of Liferay Portal map to 7.0.

App | Social Office 3.x | 7.0 | Announcements | UI enhancements | Although the Announcements app is included in 7.0, the UI enhancements from Social Office 3.x aren't. See the Announcements app's documentation for instructions on using the app in @product-ver@. | Document Library File Version Comments | Versioning Improvements | Provided as an add-on module. The *Module Installation* section below contains a link to this module. | Notifications | Various enhancements | Included | Chat | Various enhancements | This app is available for Liferay DXP and Liferay Portal CE. | Bookmarks | Various enhancements | Included | Activities | Various enhancements | Included |

Social Office Apps

Several apps were unique to Social Office. With the exception of the Tasks portlet, these apps are now in 7.0.

This table shows how apps unique to Social Office map to 7.0.

App | 7.0 | Microblogs | Included | Contacts Center | Included | Private Messaging | Provided as an add-on module. The *Module Installation* section below contains more information. | Social Office User Profile | Included | Events List | Provided as an add-on module. Click here to get it for Liferay Portal CE or here for Liferay DXP. | WYSIWYG | Provided as an add-on module. Click here to get it for Liferay Portal CE. |

UX Enhancements

The UX in 7.0 is different than previous versions. It's the first version to use Lexicon, a web implementation of Liferay's new Lexicon Experience Language. Social Office components in 7.0 use Lexicon for a consistent UX. This also means that certain Social Office components may not be where you expect them in 7.0.

This table shows where to access Social Office components in 7.0.

Component | 7.0 | Dashboard | Control Panel | Profile | Control Panel | User Bar | Control Panel | Site Navigation | My Sites app |

Here's a list of other Social Office functionality that has changed or is missing in 7.0:

- **Tasks application:** not available for 7.0. If you need it, you must upgrade its source code.
- **Site Creation Wizard:** not available for 7.0. Use site and page templates instead. An administrator should create the most commonly used site templates and let site administrators use them as an initial layout for their sites.
- **Site Navigation:** available, but modified in 7.0. Social Office's customized My Sites app let Social Office users manage their favorite sites and site memberships. You can configure 7.0's My Sites app to let users manage site memberships, but not favorites.

Great! Now you know what Social Office functionality is and isn't in 7.0. Next, you'll install a few modules in preparation for the Social Office upgrade steps.

Module Installation

To enable Social Office functionality in your 7.0 installation, you should first install a few extra modules. Deploy these modules as you would any other 7.0 module:

- **Social Office Upgrade module:** Provides a set of actions that you can run to upgrade your Social Office installation. The next section shows you how to run these actions. Click here to get this module. This module is the same for Liferay Portal CE and Liferay DXP.
- **Social Office Upgrade Association module (optional):** When the Social Office User role was granted to a user in previous versions of Liferay Portal, Social Office automatically used a site template to create public and private pages for that user. The Social Office Upgrade Association module retains this behavior (the upgrade retains the Social Office User role). Click here to get this module. This module is the same for Liferay Portal CE and Liferay DXP.
- **Document Library File Version Comments module:** By default, 7.0 users can only comment on documents in a Documents and Media repository. In Social Office, however, users could also comment on document versions. This module retains the Social Office behavior. Click here to get this module. This module is the same for Liferay Portal CE and Liferay DXP.
- **Chat:** Enables Social Office chat. This app is available for Liferay DXP and Liferay Portal CE.
- **Private Messaging:** Enables Social Office Private Messaging in 7.0. Although the first three GA releases (GA1, GA2, GA3) of Liferay Portal 7.0 CE contain this app, it will be removed in GA4 and instead released on Liferay Marketplace. To get this app for Liferay DXP, which doesn't contain it, click here.
- **Events List:** Enables Social Office Events Display in 7.0. Click here to get this app for Liferay Portal CE or here for Liferay DXP.
- **WYSIWYG:** Enables the Social Office WYSIWYG app. Click here to get this app for Liferay Portal CE or here for Liferay DXP.

Once you've installed these modules, you're ready to proceed with the upgrade. Onward!

Upgrade

Now that you're running 7.0 and the preceding modules, you're ready to upgrade your Social Office installation. As mentioned earlier, you'll do this by running a set of actions that the Social Office Upgrade modules provide.

Warning: Some of the Social Office upgrade actions may delete or alter existing data in your database. It's essential that you backup your system properly before running any of them.

You'll run these actions from the Apache Felix Gogo shell built into 7.0. With @product-ver@ running, enter the Gogo shell by running the following command in a terminal:

```
telnet localhost 11311
```

The resulting `g!` is the Gogo shell command prompt.

Warning: The Gogo shell is only available from the machine running 7.0, unless you explicitly open port 11311. For security reasons, **do not** expose the Gogo console to the outside world.

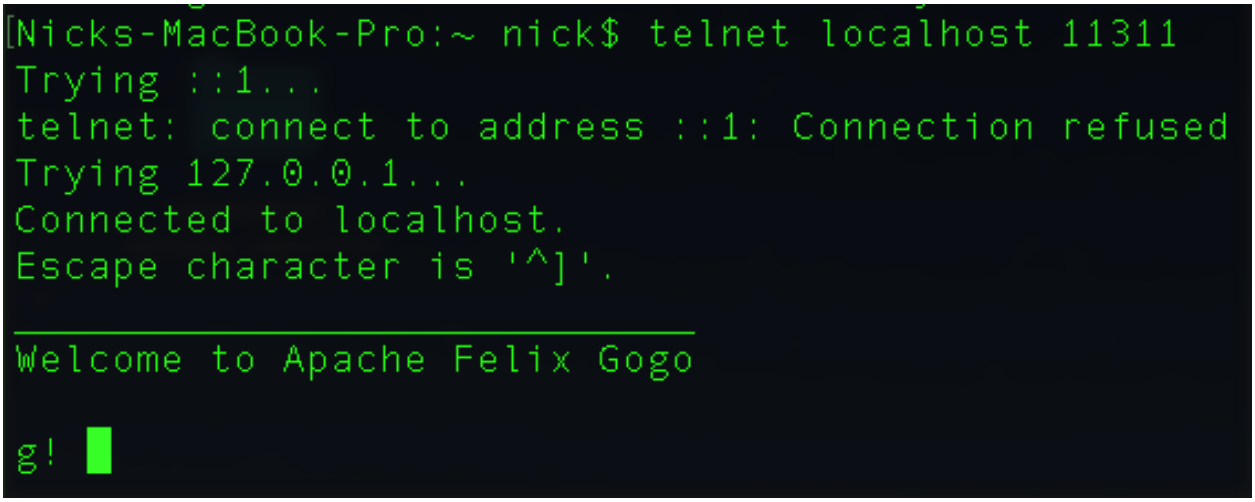


Figure 50.1: The Gogo shell lets you execute commands, including the Social Office upgrade commands, in the OSGi runtime that runs 7.0. This screenshot shows the `telnet` command that enters the shell, and the resulting Gogo shell command prompt.

Now you're ready to execute the Social Office upgrade actions. To run an action, execute it on the Gogo shell prompt. Before doing so, however, be sure to carefully read the action's description. Each action is listed here, along with its description:

- `socialOffice:executeAll`: runs all upgrade actions. Before running this action, be sure to read the descriptions for all actions.
- `socialOffice:removeTasksPortlet`: removes the Tasks portlet from all pages. Unless you manually upgrade the Tasks portlet's source code, this portlet is unavailable in 7.0. You should therefore remove it from all pages in your 7.0 installation. Note that doing so removes all references to it from all site pages; after executing this action, there's no way to restore it.
- `socialOffice:hideTasksLayout`: hides all pages that contained the Tasks portlet.
- `socialOffice:updateTheme`: restores the default theme for all pages. Unless you customized the Social Office theme, you should use the default 7.0 theme.

Next, you'll attend to some administrative tasks in your 7.0 instance.

Administration

After running the upgrade actions, there are a few administrative tasks you should complete in your 7.0 instance. These tasks help to retain Social Office functionality in 7.0.

- **Social Office User Role (optional):** The upgrade to 7.0 carries over the Social Office User role. If the optional Social Office Upgrade Association module is deployed, you can assign users to this role. Doing so gives users personal sites that function similarly to Social Office.
- **Managing Social Sites in 7.0:** The upgrade to @product-ver@ carries over all Social Office sites and site templates. To create a new social site for collaboration in 7.0, create a new site using the default Social Office site template. For more information on building sites from templates, click here.

- **My Sites:** The Site Navigation app was Social Office's custom version of the My Sites app. It was integrated into Social Office's theme. Since the Site Navigation app isn't available for 7.0, you should use My Sites instead. As mentioned earlier, however, My Sites doesn't let users manage their site memberships by default. To enable this functionality, add My Sites to a custom theme or site template that enables it. For navigating sites, users can use the new 7.0 site navigator in the Product Menu.

Chapter 51

Monitoring Liferay DXP

In this section, you will learn more about how to monitor Liferay DXP. Monitoring vital statistics such as Java memory heaps, garbage collection, database connection pools, and the application server itself means that administrators are able to optimize how Liferay DXP runs. Better monitoring means better tuning and thus avoids dangerous run time scenarios like out of memory failures and wasted heap space.

You'll learn basic monitoring techniques, such as

- Using the Visual VM tool and the JMX Console
- Garbage Collection

Read on to learn more about monitoring Liferay DXP!

51.1 Monitoring Garbage Collection and JVM

Although the tuning parameters give you a good start to tuning your JVM, you must monitor GC performance to ensure you have the best settings to meet your needs. There are several tools to help you monitor Oracle JVM performance including.

Visual VM

This tool provides a centralized console for viewing Oracle JVM performance information, including garbage collector activities.

JMX Console

This tool helps display various statistics like Liferay DXP's distributed cache performance, the performance of application server threads, JDBC connection pool usage, and more.

Note: The JMX Console is the preferred tool to use when observing Tomcat performance information.

To enable JMX connections, add these JVM arguments:

```
-Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.port=5000
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

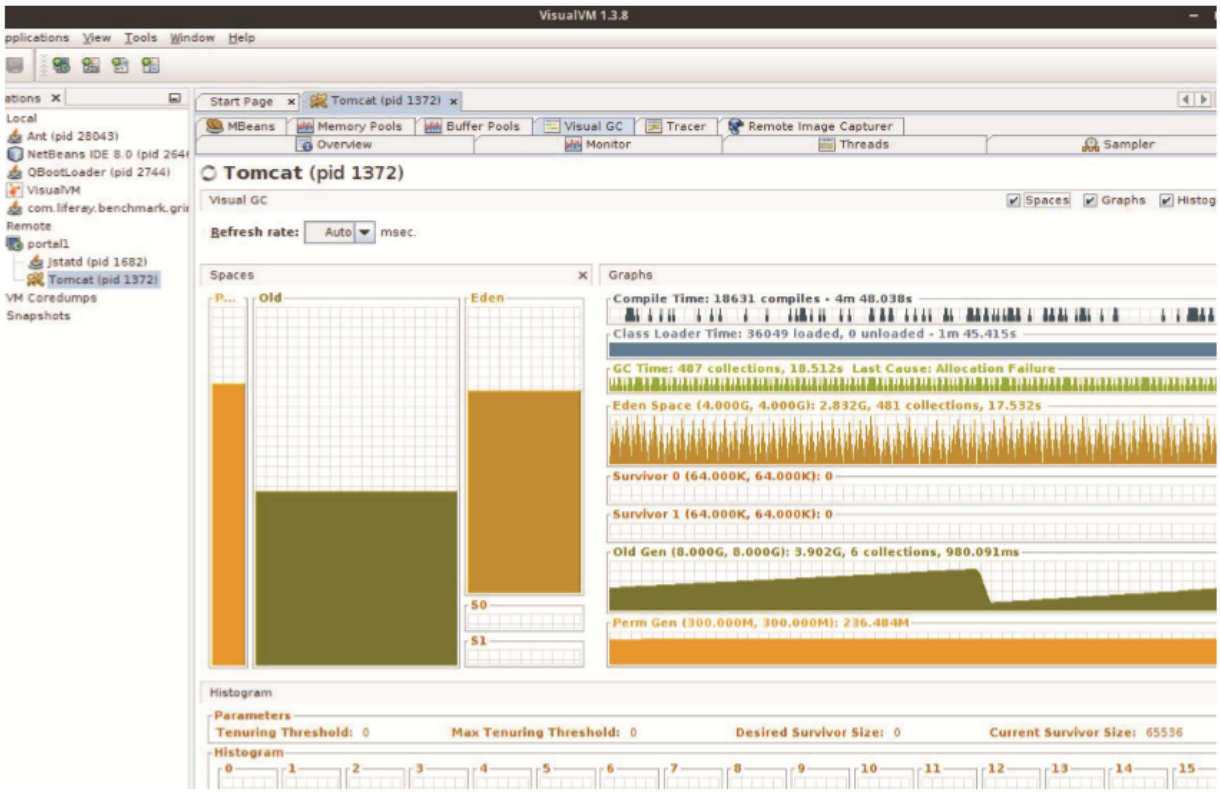


Figure 51.1: Visual VM shows the garbage collector in real-time.

If you're running JMX Console from a another machine, add these JVM arguments too:

```
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.rmi.port=5000
-Djava.rmi.server.hostname=[place IP address here]
```

Garbage Collector Verbose Logging

Add the following configuration to your JVM arguments to activate verbose logging for the JVM garbage collector.

```
-verbose:gc -Xloggc:/tmp/liferaygc1.log -XX:+PrintGCDetails
-XX:+PrintGCCause -XX:+PrintGCApplicationConcurrentTime
-XX:+PrintGCApplicationStoppedTime
```

You will need these logs to tune the JVM properly.

Note: To ensure you do have sufficient debugging information if your JVM encounters out of memory scenarios, you should consider adding this:

```
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp/dumps
```

Garbage collector log files can grow huge. Arguments like the ones below rotate the logging to a new file when the log file reaches a maximum size:

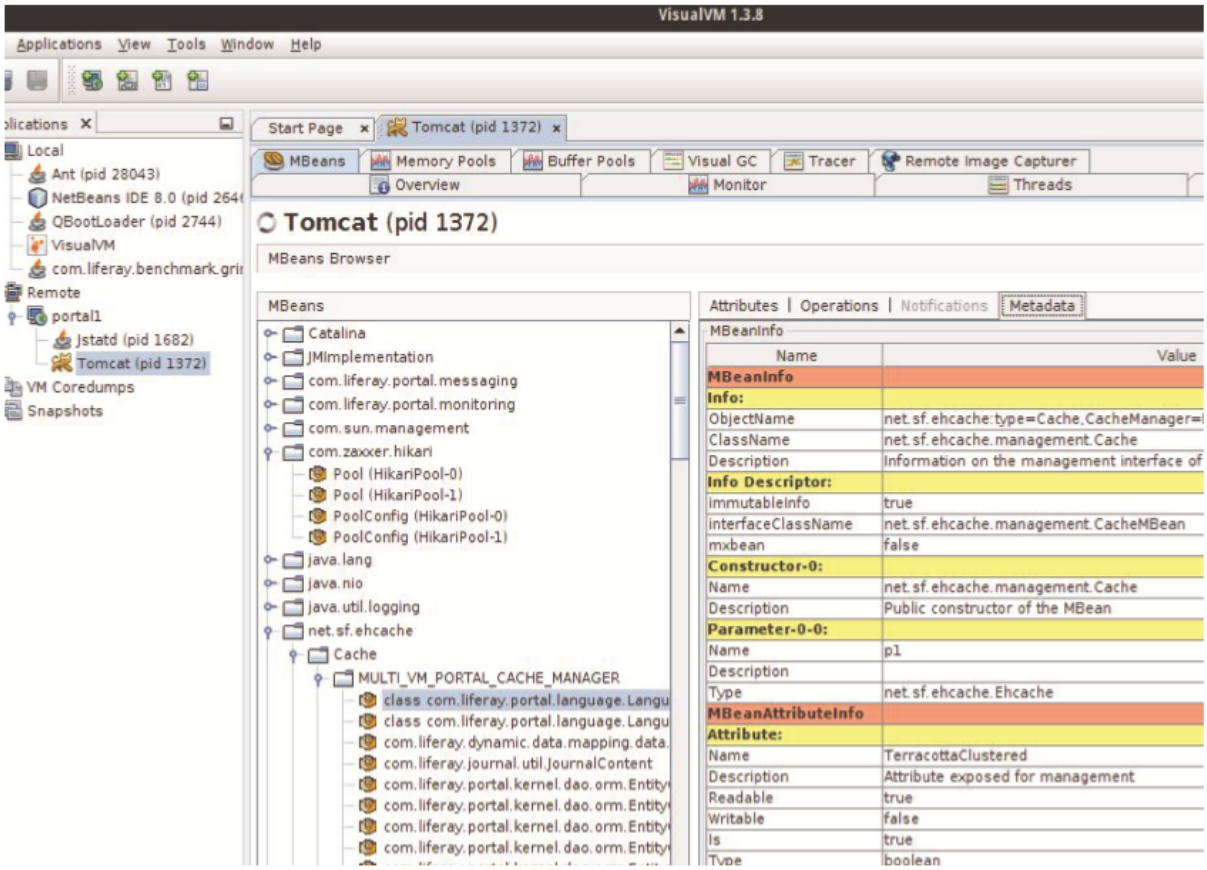


Figure 51.2: Visual VM lets you monitor using Java Management Extensions.

```
-XX:+PrintGCDateStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10
-XX:GCLogFileSize=50M
```

These arguments rotate the logs to as many as `10` log files, each with a `50M` size limit.

51.2 Advanced Monitoring: APM Tools - Dynatrace

Advanced performance monitoring tools like Dynatrace's are available for system administrators looking for more detailed information about how their Liferay DXP servers and instances are performing. Dynatrace's web-based dashboards can display real time system info on memory usage, garbage collection, CPU levels, and heap dumps. Each dashboard is its own detailed report. System administrators can view up to date information on Liferay DXP servers' performance metrics down to the individual user.

To use Dynatrace's dashboards with Liferay DXP, three things are needed: a Dynatrace client which contains the user interface (UI), a Dynatrace license, and the Dynatrace agent.

Lastly, system administrators should also sign up for a Dynatrace account if they have not already done so; this way they get access to Dynatrace's support teams.

Dashboards

Multiple dashboards are available to display ongoing transactions and processes. Because the entire Liferay DXP stack can be analyzed, there are dashboards for every component: web server, browser, application server, and database. Some of the images below are from a single instance (not clustered) of one application server (Apache Tomcat 8.0.32) connected to a MySQL 5.7 server. Installations differ but the default dashboards are the same.

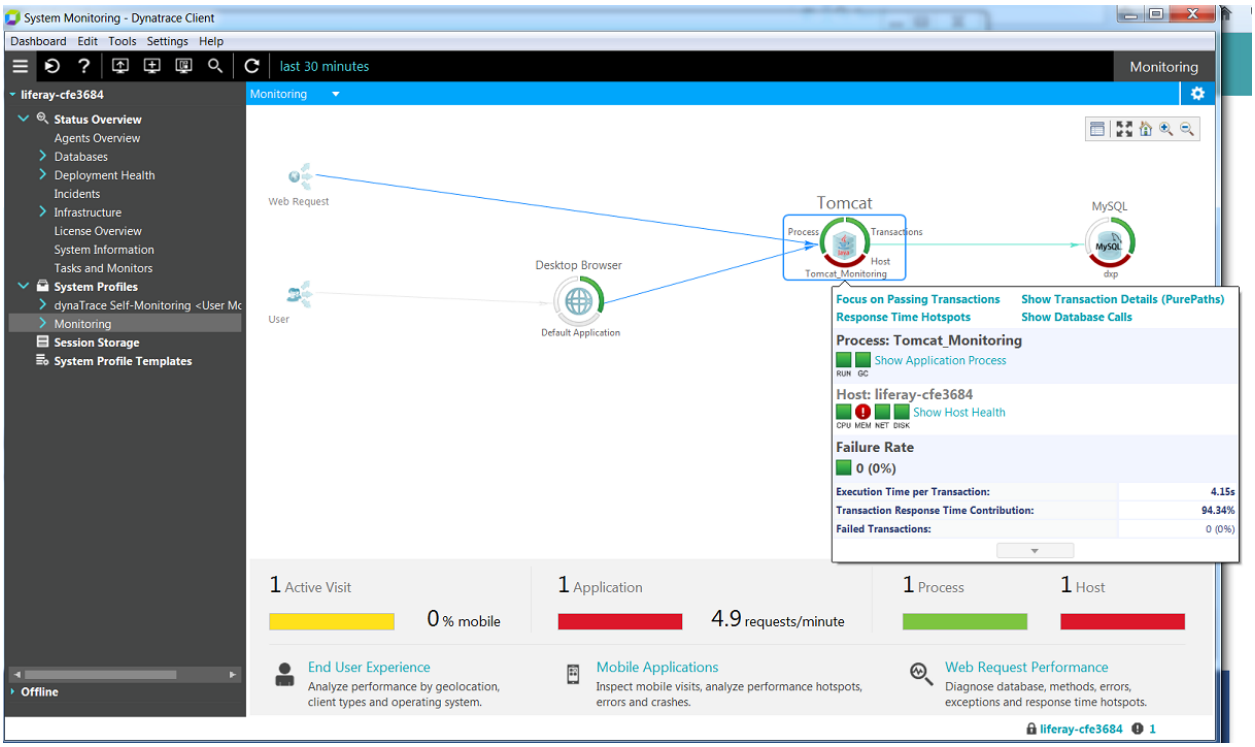


Figure 51.3: The top level gives you an interface for selecting a component to analyze.

Select *Tomcat*.

This dashboard shows the internal state of the application server. During a load test, these graphs would be much more interesting. You get a breakdown of CPU usage, memory, and more.

Returning to the root level allows administrators to view performance metrics from other parts of the environment. For example, clicking MySQL shows database transactions, which you can do without installing an agent: deploying a separate database agent is optional because the Dynatrace Collector contains database management functionality out of the box (see Database Monitoring.

One available default dashboard tracks user experience; Dynatrace can monitor user activity on a web browser or mobile app. There are three categories for each visit: Satisfied, Tolerating, and Frustrated. According to Dynatrace, a satisfied visit is one in which 1) no action failed; and 2) more than 50% of all actions were satisfied. On the other end of the spectrum, a frustrated visit is one in which 1) the last action failed (the web site does not work); and 2) the user's lastaction was frustrated (the web site was too slow). See How Does UEM Work? for more information.

If you have created a custom dashboard (usually in XML format; more information is below on the FastPack developed for Liferay), you can import it through on this menu.

1. Click the *Dashboard* menu → *Open*.

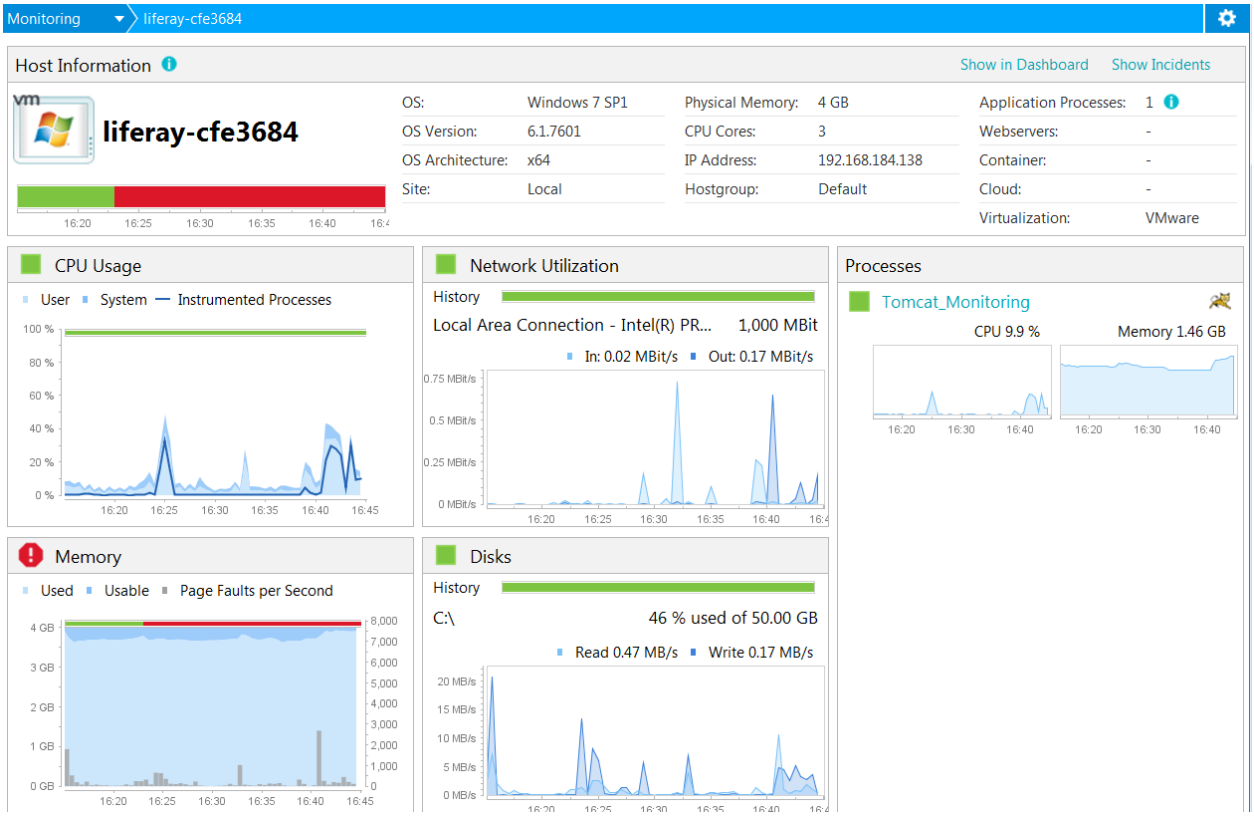


Figure 51.4: Monitoring the application server during a load gives you valuable information on your system's performance.

2. Navigate to where the dashboard has been located.

3. Click *Open Dashboard*.

Liferay Digital Enterprise FastPack

The Dynatrace FastPack for Liferay provides a pre-configured Dynatrace profile custom-tailored for Liferay Digital Enterprise 7.0 environments. It contains sensors, a template system profile with measures and business transactions, and dashboards for the Liferay DXP platform. If you're using UEM, you also get conversion and visitor tagging.

The Liferay Digital Enterprise FastPack is available for download on the Dynatrace site. These dashboards go beyond the out of the box dashboards already available through Dynatrace.

The fastpack is distributed as a `.dtp` file. To install the fastpack, follow these steps:

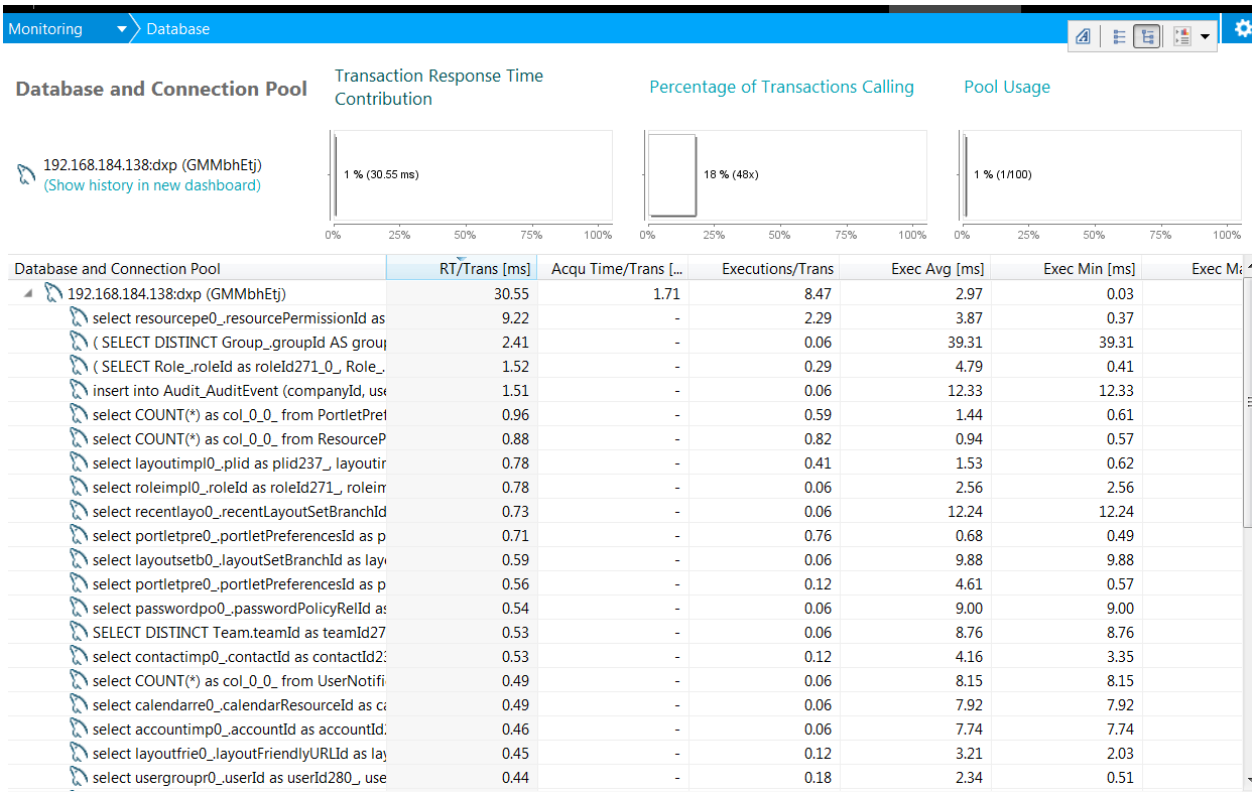
1. In the Dynatrace Client, click *Tools* → _Manage Plugins_.

2. Click *Install Plugin...*.

3. Navigate to where the `.dtp` file was downloaded.

4. Click *OK* in the *Import Resource Pack* confirmation window.

5. Click *OK* to close the *Configure Plugins* window.



Database and Connection Pool	RT/Trans [ms]	Acqu Time/Trans [...	Executions/Trans	Exec Avg [ms]	Exec Min [ms]	Exec M
192.168.184.138:dxp (GMMbhEtj)	30.55	1.71	8.47	2.97	0.03	
select resourcepe0_.resourcePermissionId as	9.22	-	2.29	3.87	0.37	
(SELECT DISTINCT Group_.groupId AS grou	2.41	-	0.06	39.31	39.31	
(SELECT Role_.roleId as roleId271_0_, Role_.	1.52	-	0.29	4.79	0.41	
insert into Audit_AuditEvent (companyId, use	1.51	-	0.06	12.33	12.33	
select COUNT(*) as col_0_0_ from PortletPref	0.96	-	0.59	1.44	0.61	
select COUNT(*) as col_0_0_ from ResourceP	0.88	-	0.82	0.94	0.57	
select layoutimpl0_.plid as plid237_, layoutir	0.78	-	0.41	1.53	0.62	
select roleimpl0_.roleId as roleId271_, roleir	0.78	-	0.06	2.56	2.56	
select recentlayo0_.recentLayoutSetBranchId	0.73	-	0.06	12.24	12.24	
select portletpre0_.portletPreferencesId as p	0.71	-	0.76	0.68	0.49	
select layoutsetb0_.layoutSetBranchId as lay	0.59	-	0.06	9.88	9.88	
select portletpre0_.portletPreferencesId as p	0.56	-	0.12	4.61	0.57	
select passwordpo0_.passwordPolicyRelId as	0.54	-	0.06	9.00	9.00	
SELECT DISTINCT Team.teamId as teamId27	0.53	-	0.06	8.76	8.76	
select contactimp0_.contactId as contactId2	0.53	-	0.12	4.16	3.35	
select COUNT(*) as col_0_0_ from UserNotifi	0.49	-	0.06	8.15	8.15	
select calendarre0_.calendarResourceId as ca	0.49	-	0.06	7.92	7.92	
select accountimp0_.accountId as accountId	0.46	-	0.06	7.74	7.74	
select layoutfrie0_.layoutFriendlyURLId as lay	0.45	-	0.12	3.21	2.03	
select usergroupr0_.userId as userId280_, use	0.44	-	0.18	2.34	0.51	

Figure 51.5: Database transactions can be viewed as they happen.

6. *Liferay* now appears in the Systems Profile left control panel.

7. Disable any other profile so that the Liferay profile is the only active profile.

Dynatrace Client Configuration

As a prerequisite to running both the latest Dynatrace Client and Liferay DXP, Java JDK 1.8 (or its equivalent) must be installed. Be sure to have enough CPU cores allocated in the JVM as well as for running Liferay DXP. Install the Fastpack plugin *after* the Client has been configured.

1. Install the Dynatrace environment. Follow the steps from the Dynatrace Installation Step 1. The installation files are available for Windows, Unix, and Linux systems.

   Windows users, download then install the full `.msi` file (approx. 770MB).

   I. Be sure to check *Immediately activate the Dynatrace .NET agent*.

   II. Be sure to check *Immediately activate the Dynatrace IIS agent*.
   III. Be sure to start all three: Dynatrace Server, Collector, and Client

   Linux users, the installation files are packaged as executable jar. Download the full jar. Run the command `java -jar dynatrace-full-linux-x86-64.jar` wherever the jar has been downloaded.

2. Start the Dynatrace instance and deploy the Dynatrace trial license (see below).

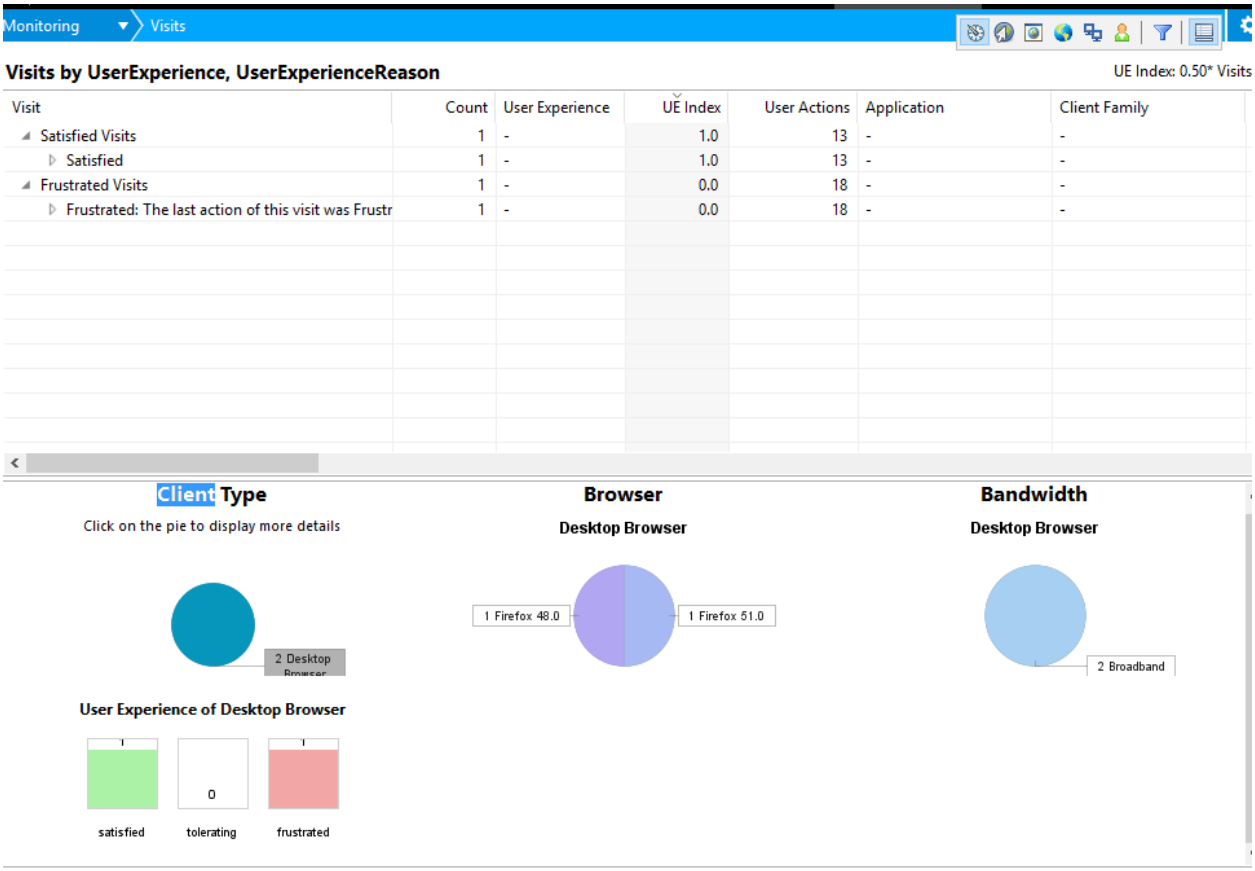


Figure 51.6: Browser metrics help you find client-side performance problems.

a. Request the Dynatrace license key; the key is available either through your web credentials or from a link in the welcome email. Place the file in a suitable location. When the client starts for the first time and prompts for the license, navigate to the file.

b. Once the license has been successfully imported, the Dynatrace Client will prompt for a server restart.

3. After the server restarts, verify that the following services have started:

   - Dynatrace Server
   - Dynatrace Front-End
   - Dynatrace Collector

   You can do this by checking to see if the processes are running:

```
# ps -A | grep dt
```

   You should see output like this:

```
3954 ?00:00:43 dtcollector
5924 ?00:01:54 dtserver
5949 ?00:00:42 dtfrontendserver
```

Use netstat to make sure the client is listening on the proper ports:

```
# netstat -an | grep 021
```

You should see output like this:

```
 tcp  0  0 :::2021 :::* LISTEN (port for dynaTrace client connections)
 tcp  0  0 :::8021 :::* LISTEN (the dynaTrace server's web interface)
```

```
# netstat -an | grep 99
```

You should see output like this:

```
tcp        0      0 :::6699     :::*     LISTEN    (port for dynaTrace collector connections)
tcp        0      0 :::9998     :::*     LISTEN    (port for dynaTrace agent connections)
```

Congratulations! You have the Dynatrace client installed!

Dynatrace Agent Configuration

Deploy the Dynatrace agent to the Liferay DXP servers. The agent sits on top of existing infrastructure (in this case, Apache Tomcat).

1. Download the agent `.jar` for your operating system. Use this agent only if the Dynatrace Client and the Liferay DXP bundle on Tomcat are connected remotely. Administrators must install the agent on the same machine Liferay DXP is on.

2. The location of the agent must be set in each application server. If the client and Liferay DXP are on the same machine, you don't have to install anything else (the agent is already included in the full installation), but you do have to configure it. This configuration is only for testing and demonstration purposes; the Dynatrace Client's JVM requirements can be different from Liferay DXP's. Either way, edit the setenv.bat|sh:

   ```
   `-agentpath:${location of the dtagent .dll}"=name=Tomcat_Monitoring,server=liferay-cfe3684:9998`
   ```

   For example in Windows:

   ```
   -agentpath:"C:\Program Files\Dynatrace\Dynatrace 6.5\agent\lib64\dtagent.dll"=name=Tomcat_Monitoring,server=liferay-
   cfe3684:9998
   ```

   For other application servers, place the property in the file where other JVM settings are set (for example, `standalone.conf.bat` for JBoss EAP).

3. Connect the Liferay DXP with the Dynatrace agent to the Dynatrace instance. Refer to Dynatrace's documentation for further details. Click the Application Servers tab on the right. The example below uses Apache Tomcat but according to Dynatrace, the steps are virtually the same for JBoss and WebSphere.

   a. Select *Client → Monitoring*.
   b. Select *Java*.
   c. Select *App Server*.

d. Select *Apache 5+*.
e. Select *Java 5 or later (64 bit)*.
f. Select whether the Tomcat bundle is *local* or *remote*.
g. As long as the connector is on, it should find the Liferay DXP instance. For testing purposes, the Liferay DXP bundle is local.

4. After you import the fast pack, you must

   a. Select the Liferay profile as the only active system profile in Dynatrace, and
   b. Restart the application server.

5. Then your Liferay agent starts sending data into the newly imported profile.

Performance Testing

Systems administrators must often perform load testing as part of an overall tuning process. The Dynatrace Client is a great tool to monitor and then analyze the performance of your system. For demonstration purposes, JMeter was used to create a simple load test, and the Dynatrace Client captured the results illustrated below.

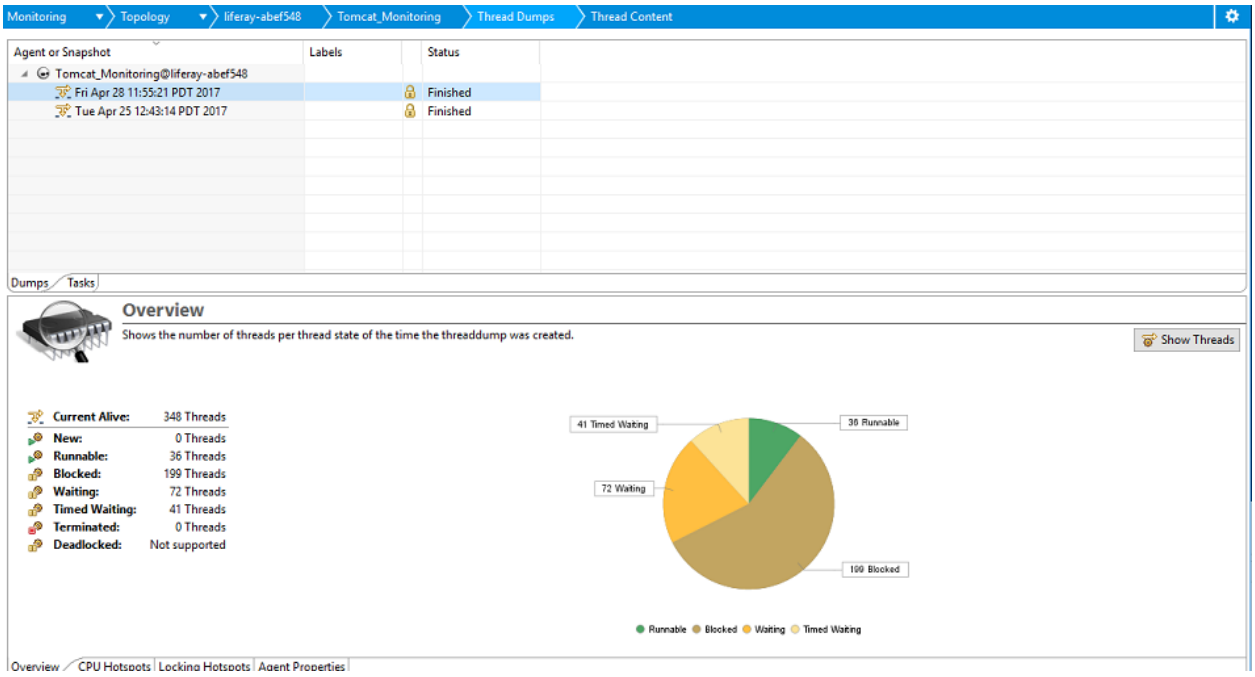


Figure 51.7: Dynatrace Client can generate thread dump reports.

The Dynatrace Client shows garbage collection over time as the number of threads increase during the load test.

Here, the Dynatrace Client continues to display in real time the high consumption during the load test.

One more dashboard of note during a performance test is the CPU Sampling dashboard. The image below is a report generated to capture the CPU process over 100 seconds. This is helpful to show administrators unwanted processes slowing down a Liferay DXP instance.

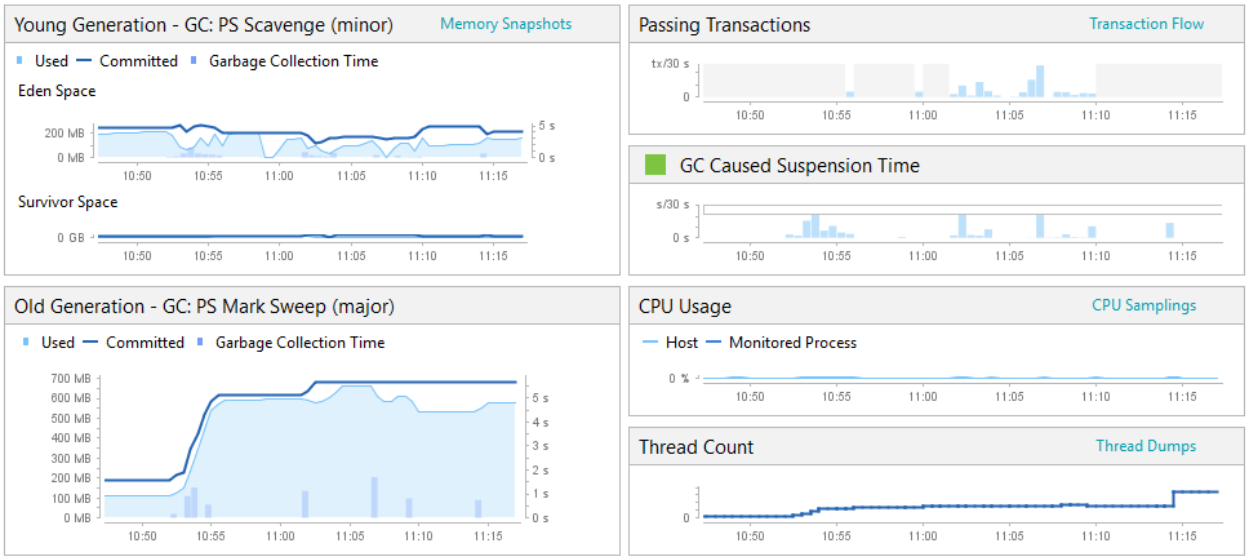


Figure 51.8: Garbage collection statistics at the beginning of the load test.

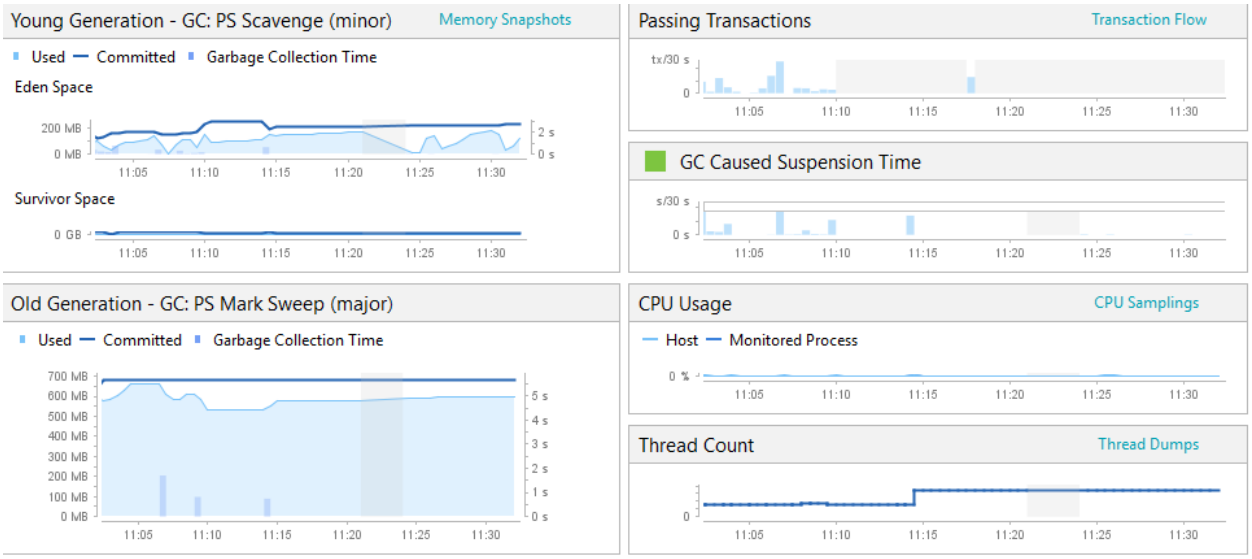
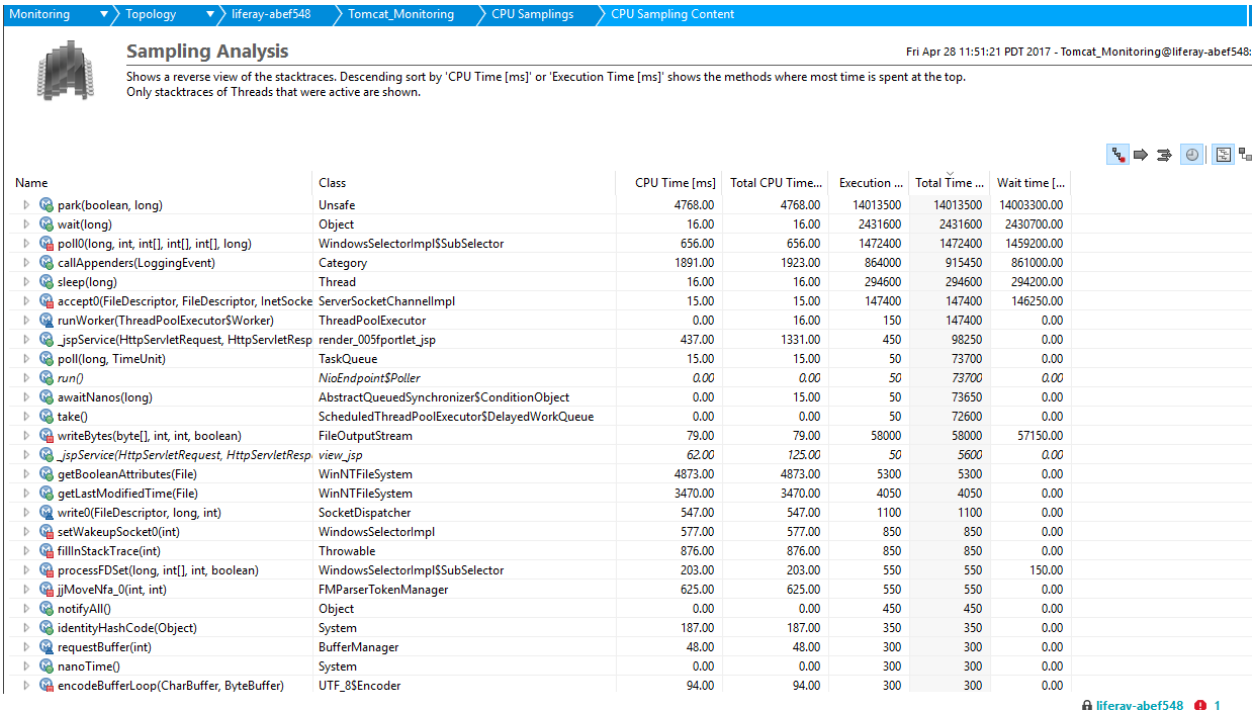


Figure 51.9: Garbage collection statistics at the middle of the load test.



Name	Class	CPU Time [ms]	Total CPU Time...	Execution ...	Total Time ...	Wait time [...
park(boolean, long)	Unsafe	4768.00	4768.00	14013500	14013500	14003300.00
wait(long)	Object	16.00	16.00	2431600	2431600	2430700.00
poll0(long, int, int[], int[], int[], long)	WindowsSelectorImpl$SubSelector	656.00	656.00	1472400	1472400	1459200.00
callAppenders(LoggingEvent)	Category	1891.00	1923.00	864000	915450	861000.00
sleep(long)	Thread	16.00	16.00	294600	294600	294200.00
accept0(FileDescriptor, FileDescriptor, InetSocke	ServerSocketChannelImpl	15.00	15.00	147400	147400	146250.00
runWorker(ThreadPoolExecutor$Worker)	ThreadPoolExecutor	0.00	16.00	150	147400	0.00
_jspService(HttpServletRequest, HttpServletResp	render_005fportlet_jsp	437.00	1331.00	450	98250	0.00
poll(long, TimeUnit)	TaskQueue	15.00	15.00	50	73700	0.00
run()	*NioEndpoint$Poller*	*0.00*	*0.00*	*50*	*73700*	*0.00*
awaitNanos(long)	AbstractQueuedSynchronizer$ConditionObject	0.00	15.00	50	73650	0.00
take()	ScheduledThreadPoolExecutor$DelayedWorkQueue	0.00	0.00	50	72600	0.00
writeBytes(byte[], int, int, boolean)	FileOutputStream	79.00	79.00	58000	58000	57150.00
_jspService(HttpServletRequest, HttpServletResp	*view_jsp*	*62.00*	*125.00*	*50*	*5600*	*0.00*
getBooleanAttributes(File)	WinNTFileSystem	4873.00	4873.00	5300	5300	0.00
getLastModifiedTime(File)	WinNTFileSystem	3470.00	3470.00	4050	4050	0.00
write0(FileDescriptor, long, int)	SocketDispatcher	547.00	547.00	1100	1100	0.00
setWakeupSocket0(int)	WindowsSelectorImpl	577.00	577.00	850	850	0.00
fillInStackTrace(int)	Throwable	876.00	876.00	850	850	0.00
processFDSet(long, int[], int, boolean)	WindowsSelectorImpl$SubSelector	203.00	203.00	550	550	150.00
jjMoveNfa_0(int, int)	FMParserTokenManager	625.00	625.00	550	550	0.00
notifyAll()	Object	0.00	0.00	450	450	0.00
identityHashCode(Object)	System	187.00	187.00	350	350	0.00
requestBuffer(int)	BufferManager	48.00	48.00	300	300	0.00
nanoTime()	System	0.00	0.00	300	300	0.00
encodeBufferLoop(CharBuffer, ByteBuffer)	UTF_8$Encoder	94.00	94.00	300	300	0.00

Figure 51.10: Dynatrace Client can generate a CPU Sampling report.

All these dashboards come out-of-the-box, and even more dashboards are available after deploying the Dynatrace FastPack developed for Liferay DXP. Using Dynatrace provides many advantages for performance monitoring.

Resources

1. Dynatrace Community
2. Dynatrace Installation Step 1
3. Dynatrace Installation Step 2
4. Dynatrace Installation Step 3

Chapter 52

Liferay Enterprise Search

Search engines are a critical component of your Liferay DXP installation. They allow you to provide documents in an index that can quickly be searched, rather than relying on expensive database queries. For more fundamental material on search and indexing in Liferay DXP, read here. If you're here, you probably know the basics already and want to get started on configuring a search engine for your 7.0 deployment.

Liferay DXP ships with Elasticsearch, a highly scalable, full-text search engine. Elasticsearch is well supported and will almost certainly meet any search and indexing need you have. For deployment settings, learn to configure a standalone, or remote, Elasticsearch server or cluster here.

If you need to use Solr, it's also supported in Liferay DXP, and you can read more about its configuration here.

52.1 Preparing to Install Elasticsearch

By default, 7.0 and its embedded Elasticsearch engine run in the same JVM. Although this enables out-of-the-box search in 7.0, it's only supported for development; production use isn't supported. For production use, Liferay only supports Elasticsearch when it runs in a separate JVM, because search engines benefit heavily from caching. This makes their JVM memory profiles differ substantially from those of a JVM running 7.0. Therefore, the two applications should always be kept separate in production environments.

The following sections provide a synopsis of Elasticsearch configurations for 7.0. Prior to deployment, we strongly recommend reading Elastic's documentation on production deployment.

Sizing Your Deployment

When sizing your Elasticsearch deployment, you must carefully consider your CPU, memory, disk, and network capacity. Generally, you should deploy Elasticsearch on medium to large machines. This lets you scale effectively and avoid large numbers of machines. You should also avoid running multiple Elasticsearch JVMs on the same operating system.

CPU

Liferay recommends that you allocate at least 4 CPU cores to the Elasticsearch engine. This assumes only 1 Elasticsearch JVM running on the machine.

Memory

Liferay recommends at least 16 GB of memory, with 64 GB preferred. The memory allocation depends upon the amount of index data. For index sizes 500 GB to 1 TB, 64 GB of memory should suffice.

Disk

Search engines store their indexes on disk. Disk I/O capacity can therefore impact search performance. Liferay recommends deploying Elasticsearch on SSD when possible. If you are unable to do this, use high-performance traditional hard disks (e.g., 15k RPM). Consider using RAID 0 for both SSD and traditional hard disks.

In general, avoid using NAS (network attached storage) for Elasticsearch as the network overhead can be large. If you're using public cloud infrastructure like Amazon Web Services, use instance local storage and avoid network storage like Elastic Block Store (EBS).

Also note that search index sizes vary based on the indexed content. Ensure you have at least 25% more disk capacity than the total size of your indexes. For example, if your index is 50 GB, you should have at least 75 GB of disk space available. To estimate the disk space you need, Liferay recommends that you index a representative sample of your production content and then multiply the size of that sample index by the fraction of your production content that it represents. For example, index 25% of your production content and then multiply the resulting index size by 4. Keep in mind that indexing a 1 MB file doesn't result in 1 MB of space in the search index.

Networking

Elasticsearch relies on clustering and sharding to deliver fast, accurate search results. Therefore, it requires a fast and reliable network. Most modern data centers provide 1 GbE or 10 GbE between machines. Avoid spreading Elasticsearch clusters across multiple data centers. Elasticsearch doesn't support multi-data center deployments, especially data centers spread across large distances (e.g., cross-continent). To support multi-data center deployments, you must create a custom solution that distributes index requests (update, delete, add document) to each data center.

Shards and Replicas

Elasticsearch uses shards and replicas to scale. Shards divide a search index into smaller, more manageable chunks. For example, if you have a 500 GB index you can split it into 10 shards of 50 GB each. For best results, each shard shouldn't exceed 50 GB. More shards generally mean faster indexing (write) performance but slower search (read) performance. In Elasticsearch, an index with multiple shards results in a distributed search and a subsequent result merge. Replicas provide resiliency and improve search performance. A replica helps to load balance search operations across the cluster.

52.2 Configuring Elasticsearch

Liferay DXP is an open source project, so you won't be surprised to learn that its default search engine is also an open source project. Elasticsearch is a highly scalable, full-text search and analytics engine.

By default, Elasticsearch runs as an embedded search engine, which is useful for development and testing but is not supported in production. In production environments you must run Elasticsearch in remote mode, as a separate server or cluster. This guide walks you through the process of configuring Elasticsearch in remote mode.

If you'd rather use Solr, it's also supported. See here for information on installing and configuring Solr.

To get up and running quickly with Elasticsearch as a remote server, refer to the Installing Elasticsearch article. In that article you'll find the basic instructions for the installation and configuration of Elasticsearch in a single server environment. This article includes more details and information on clustering and tuning Elasticsearch. In this article you'll learn to configure your existing Elasticsearch installation for use in production environments.

If you've come here looking for information on search engines in general, or the low level search infrastructure of Liferay DXP, refer instead to the developer tutorial Introduction to Liferay Search.

These terms will be useful to understand as you read this guide:

- *Elasticsearch Home* refers to the root folder of your unzipped Elasticsearch installation (for example, `elasticsearch-2.4.0`).
- *Liferay Home* refers to the root folder of your Liferay DXP installation. It contains the `osgi`, `deploy`, `data`, and `license` folders, among others.

Upgrading to Elasticsearch 6.5: If you have an existing 2.4 or 6.1 Elasticsearch installation, follow the upgrade guide to move onto Elasticsearch 6.5.

Embedded vs. Remote Operation Mode

When you install Liferay DXP, there's an embedded Elasticsearch already installed. In embedded mode, Elasticsearch search runs in the same JVM to make it easy to test-drive with minimal configuration. Running both servers in the same process has drawbacks:

- Elasticsearch must use the same JVM options as Liferay DXP.
- Liferay DXP and Elasticsearch compete for resources.

You wouldn't run an embedded database like HSQL in production, and you shouldn't run Elasticsearch in embedded mode in production either. Instead, run Elasticsearch in *remote operation mode*, as a standalone server or cluster of server nodes.

Configuring Elasticsearch

For detailed Elasticsearch configuration information, refer to the Elasticsearch documentation.

The name of your Elasticsearch cluster is important. When you're running Elasticsearch in remote mode, the cluster name is used by Liferay DXP to recognize the Elasticsearch cluster. To learn about setting the Elasticsearch cluster name on the Liferay DXP side, refer below to the section called Configuring the Liferay Elasticsearch adapter.

Elasticsearch's configuration files are written in YAML and kept in the `[Elasticsearch Home]/config` folder:

- `elasticsearch.yml` is for configuring Elasticsearch modules
- `logging.yml` is for configuring Elasticsearch logging

To set the name of the Elasticsearch cluster, open `[Elasticsearch Home]/config/elasticsearch.yml` and specify

```
cluster.name: LiferayElasticsearchCluster
```

Since `LiferayElasticsearchCluster` is the default name given to the cluster in Liferay DXP, this would work just fine. Of course, you can name your cluster whatever you'd like (we humbly submit the recommendation `clustery_mcclusterface`).1 You can configure your node name using the same syntax (setting the `node.name` property).

If you'd rather work from the command line than in the configuration file, navigate to Elasticsearch Home and enter

```
./bin/elasticsearch --cluster.name clustery_mcclusterface --node.name nody_mcnodeface
```

Feel free to change the node name or the cluster name. Once you configure Elasticsearch to your liking, start it up.

Starting Elasticsearch

Start Elasticsearch by navigating to Elasticsearch Home and typing

```
./bin/elasticsearch
```

if you run Linux, or

```
\bin\elasticsearch.bat
```

if you run Windows.

To run as a daemon in the background, add the `-d` switch to either command:

```
./bin/elasticsearch -d
```

When you have Elasticsearch itself installed and running, and Liferay DXP installed and running (do that if you haven't already) you need to introduce Liferay DXP and Elasticsearch to each other. Fortunately, Liferay provides an adapter that helps it find and integrate your Elasticsearch cluster.

Configuring the Liferay Elasticsearch Adapter

The Elasticsearch connector is a module that ships with the Foundation Suite and deployed to the OSGi runtime, titled *Liferay Portal Search Elasticsearch*. This connector provides integration between Elasticsearch and Liferay DXP. Before you configure the adapter, make sure Elasticsearch is running.

Elasticsearch 6.5: The connector for Elasticsearch 6.5 is called *Liferay Connector to Elasticsearch 6*. Download the Elasticsearch 6.5 connector from Liferay Marketplace and install it, following the Marketplace documentation if necessary.

There are two ways to configure the adapter:

1. Use the System Settings application in the Control Panel.
2. Manually create an OSGi configuration file.

It's convenient to configure the Elasticsearch adapter from System Settings, but this is often only possible during development and testing. If you're not familiar with System Settings, you can read about it here. Even if you need a configuration file so you can use the same configuration on another Liferay DXP system, you can still use System Settings. Just make the configuration edits you need, then export the `.config` file with your configuration.

Configuring the Adapter in the Control Panel

Here are the steps to configure the Elasticsearch adapter from the System Settings application:

1. Start Liferay DXP.
2. Navigate to *Control Panel* → *Configuration* → *System Settings* → *Foundation*.
3. Find the *Elasticsearch* entry (scroll down and browse to it or use the search box) and click the Actions icon (⋮), then *Edit*.

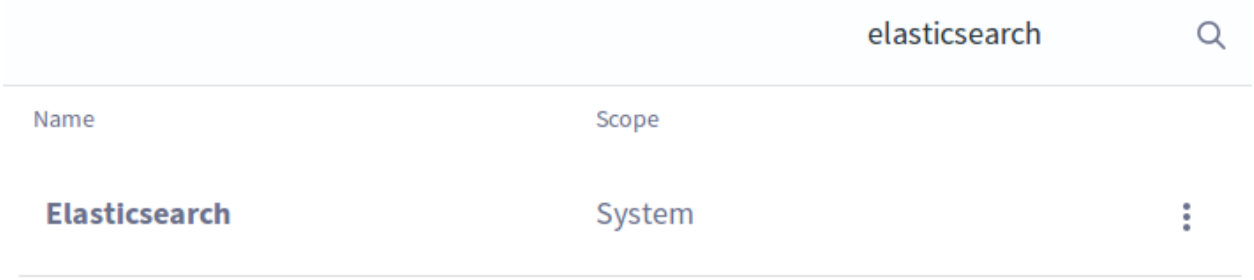


Figure 52.1: Use the System Settings application in Liferay DXP's Control Panel to configure the Elasticsearch adapter.

4. Change Operation Mode to *Remote*, and then click *Save*.
5. After you switch operation modes (`EMBEDDED` → `REMOTE`), you must trigger a re-index. Navigate to *Control Panel* → *Server Administration*, find the *Index Actions* section, and click *Execute* next to *Reindex all search indexes*.

Configuring the Adapter with an OSGi `.config` File

When preparing a system for production deployment, you want to set up a repeatable deployment process. Therefore, it's best to use the OSGi configuration file, where your configuration is maintained in a controlled source.

Follow these steps to configure the Elasticsearch adapter using an OSGi configuration file:

1. Create the following file to configure the default adapter (for Elasticsearch 2.4):

```
[Liferay_Home]/osgi/configs/com.liferay.portal.search.elasticsearch.configuration.ElasticsearchConfiguration.config
```

   To configure the Liferay Connector to Elasticsearch 6, name your file thus:

```
[Liferay_Home]/osgi/configs/com.liferay.portal.search.elasticsearch6.configuration.ElasticsearchConfiguration.config
```

2. Add this to the configuration file you just created:

```
operationMode="REMOTE"
# If running Elasticsearch from a different computer:
#transportAddresses="ip.of.elasticsearch.node:9300"
# Highly recommended for all non-prodcution usage (e.g., practice, tests, diagnostics):
#logExceptionsOnly="false"
```

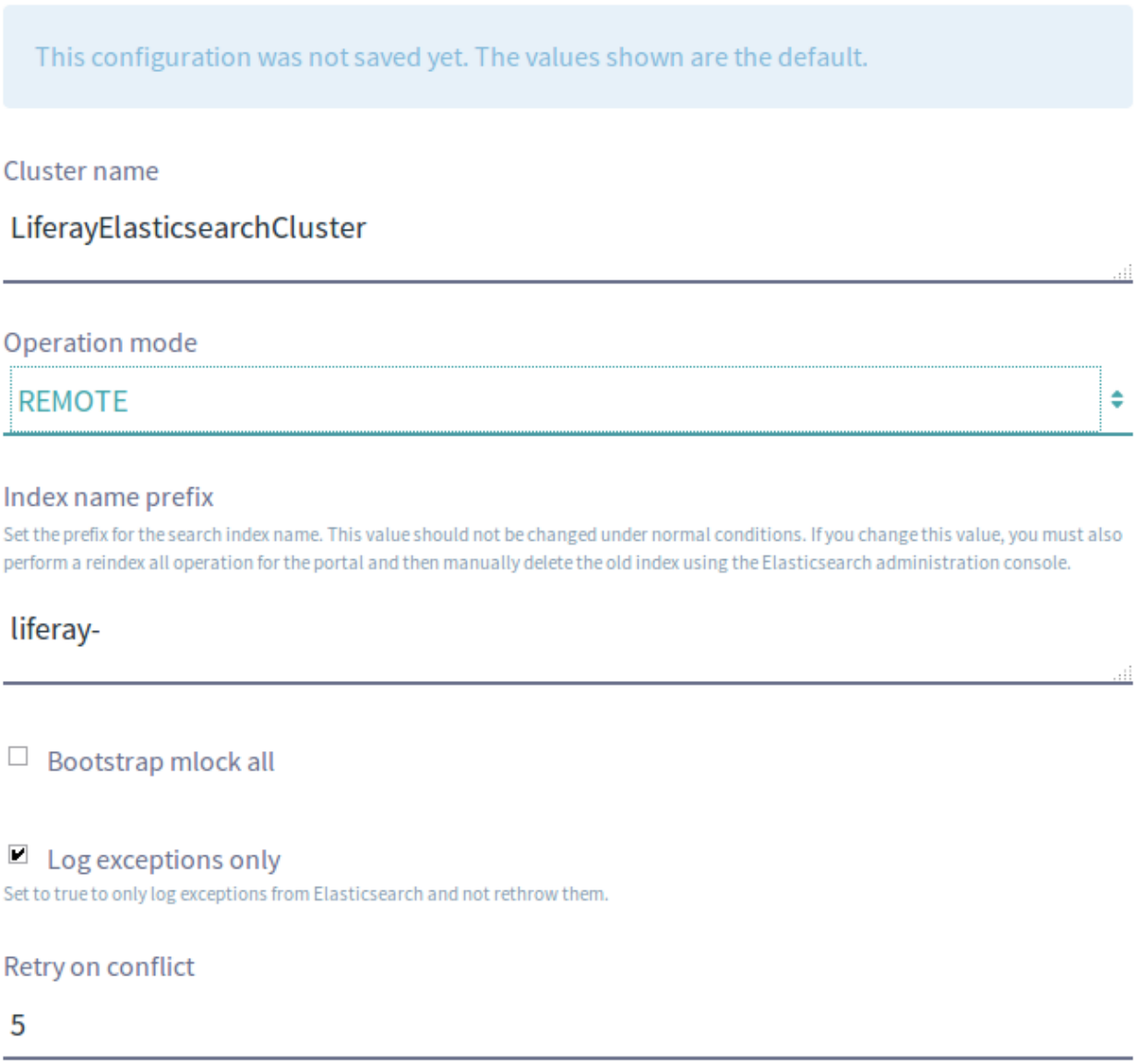


Figure 52.2: Set Operation Mode to *Remote* from System Settings.

3. Start Liferay DXP or re-index if already running.

As you can see from the System Settings entry for Elasticsearch, there are a lot more configuration options available that help you tune your system for optimal performance. For a detailed accounting of these, refer to the reference article on Elasticsearch Settings.

What follows here are some known good configurations for clustering Elasticsearch. These, however, can't replace the manual process of tuning, testing under load, and tuning again, so we encourage you to examine the settings as well as the Elasticsearch documentation and go through that process once you have a working configuration.

Configuring a Remote Elasticsearch Host

In production systems Elasticsearch and Liferay DXP are installed on different servers. To make Liferay DXP aware of the Elasticsearch cluster, set

```
transportAddresses=[IP address of Elasticsearch Node]:9300
```

in the Elasticsearch adapter's OSGi configuration file. List as many or as few Elasticsearch nodes in this property as you'd like. This tells Liferay DXP the IP address or host name where search requests are to be sent. If using System Settings, set the value in the *Transport Addresses* property.

Note: In an Elasticsearch cluster you can list the transport addresses for multiple Elasticsearch nodes, if appropriate. Just use a comma-separated list in the `transportAddresses` property. If you set only one transport address, Liferay DXP loses contact with Elasticsearch if that node goes down.

On the Elasticsearch side, set the `network.host` property in your `elasticsearch.yml` file. This property simultaneously sets both the *bind host* (the host Elasticsearch listens on for requests) and the *publish host* (the host name or IP address Elasticsearch uses to communicate with other nodes). See here for more information.

Clustering Elasticsearch in Remote Operation Mode

Clustering Elasticsearch is easy. Each time you run the Elasticsearch start script, a new local storage node is added to the cluster. If you want four nodes running locally, for example, just run `./bin/elasticsearch` four times. If you only run the start script once, you have a cluster with just one node.

Elasticsearch 6.5: To start multiple local storage nodes in Elasticsearch 6.5, you must also configure `node.max_local_storage_nodes` to be something greater than `1`. See here for more information.

Elasticsearch's default configuration works for a cluster of up to ten nodes, since the default number of shards is `5`, while the default number of replica shards is `1`:

```
index.number_of_shards: 5
index.number_of_replicas: 1
```

Note: Elasticsearch uses the Zen Discovery Module by default, which provides unicast discovery. Additionally, nodes in the cluster communicate using the Transport Module, through TCP. See the Elasticsearch documentation for the available properties (to be set in the `elasticsearch.yml` file), and the Liferay DXP Elasticsearch Adapter's reference article for the adapter's available settings.

At a minimum, provide the list of hosts to act as gossip routers during unicast discovery in the `elasticsearch.yml`:

```
discovery.zen.ping.unicast.hosts: ["node1.ip.address", "node2.ip.address"]
```

Elasticsearch 6.5: Elasticsearch 6 removed the setting that specifies the number of shards and replicas in the `elasticsearch.yml`file. Configure these index-level settings in the Elasticsearch 6.5 connector, using the `indexNumberOfShards` and `indexNumberOfReplicas` properties to specify the number of primary shards and number of replica shards, respectively.

For more information on configuring an Elasticsearch cluster, see the documentation on Elasticsearch Index Settings.

Advanced Configuration of the Liferay Elasticsearch Adapter

The default configurations for Liferay's Elasticsearch adapter module are set in a Java class called `ElasticsearchConfiguration`.

While the Elasticsearch adapter has a lot of configuration options out of the box, you might find an Elasticsearch configuration you need that isn't provided by default. In this case, add the configuration options you need. If something is configurable for Elasticsearch, its configurable using the Elasticsearch adapter.

Adding Settings and Mappings to the Liferay Elasticsearch Adapter

The available configuration options are divided into two groups: the ones you'll use most often by default, and a catch-all for everything else. So if the necessary setting isn't available by default, you can still configure it with the Liferay Elasticsearch adapter. Just specify the settings you need by using one or more of the `additionalConfigurations`, `additionalIndexConfigurations`, or `additionalTypeMappings` settings.

Adding Configurations

`additionalConfigurations` is used to define extra settings (defined in YAML) for the embedded Elasticsearch or the local Elasticsearch client when running in remote mode. In production, only one additional configuration can be added here:

```
client.transport.ping_timeout
```

Elasticsearch 6.5: The Elasticsearch 6.5 connector includes the `client.transport.ping_timeout` as a native setting. Configure it through its dedicated setting rather than with `additionalConfigurations`.

The rest of the settings for the client are available as default configuration options in the Liferay Elasticsearch adapter. See the Elasticsearch Settings reference article for more information. See the Elasticsearch documentation for a description of all the client settings and for an example.

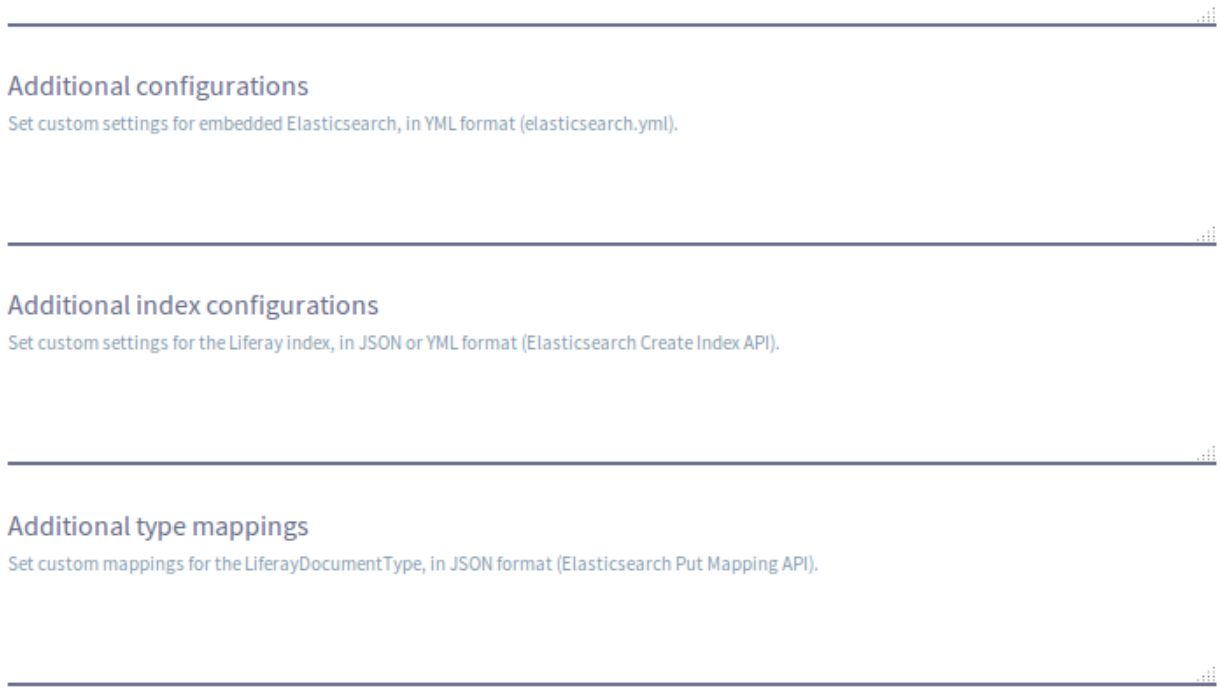


Figure 52.3: You can add Elasticsearch configurations to the ones currently available in System Settings.

Adding Index Configurations

`additionalIndexConfigurations` is used to define extra settings (in JSON or YAML format) that are applied to the Liferay DXP index when it's created. For example, you can create custom analyzers and filters using this setting. For a complete list of available settings, see the Elasticsearch reference.

Here's an example that shows how to configure analysis that can be applied to a field or dynamic template (see below.

```
{
    "analysis": {
        "analyzer": {
            "kuromoji_liferay_custom": {
                "filter": [
                    "cjk_width",
                    "kuromoji_baseform",
                    "pos_filter"
                ],
                "tokenizer": "kuromoji_tokenizer"
            }
        },
        "filter": {
            "pos_filter": {
                "type": "kuromoji_part_of_speech"
            }
        }
    }
}
```

Adding Type Mappings

`additionalTypeMappings` is used to define extra mappings for the `LiferayDocumentType` type definition, which are applied when the index is created. Add mappings using JSON syntax. For more information see here and here. Use `additionalTypeMappings` for new field (`properties`) and dynamic template mappings, but do not try to override existing mappings. If any of the mappings set here overlap with existing mappings, index creation will fail. Use `overrideTypeMappings` to replace the default mappings.

As with dynamic templates, you can add sub-field mappings to Liferay DXP's type mapping. These are referred to as properties in Elasticsearch.

```
{
    "LiferayDocumentType": {
        "properties": {
            "fooName": {
                "index": "not_analyzed",
                "store": "yes",
                "type": "string"
            }
        }
    }
}
```

Elasticsearch 6: The above property mapping looks different in Elasticsearch 6.1:

```
{
    "LiferayDocumentType": {
        "properties": {
            "fooName": {
                "index": "true",
                "store": "true",
                "type": "keyword"
            }
        }
    }
}
```

See here for more details on Elasticsearch's field datatypes.

The above example shows how a `fooName` field might be added to Liferay DXP's type mapping. Because `fooName` is not an existing property in the mapping, it will work just fine. If you try to override an existing property mapping, index creation will fail. Instead use the `overrideTypeMappings` setting to override `properties` in the mapping.

To see that your additional mappings have been added to the `LiferayDocumentType`, navigate to this URL after saving your additions and reindexing:

```
http://[HOST]:[ES_PORT]/liferay-[COMPANY_ID]/_mapping/LiferayDocumentType?pretty
```

Here's what it would look like for an Elasticsearch instance running on `localhost:9200`, with a Liferay DXP Company ID of `20116`:

```
http://localhost:9200/liferay-20116/_mapping/LiferayDocumentType?pretty
```

In the above URL, `liferay-20116`is the index name. Including it indicates that you want to see the mappings that were used to create the index with that name.

Overriding Type Mappings

Use `overrideTypeMappings` to override Liferay DXP's default type mappings. This is an advanced feature that should be used only if strictly necessary. If you set this value, the default mappings used to define the Liferay Document Type in Liferay DXP source code (for example, `liferay-type-mappings.json`) are ignored entirely, so include the whole mappings definition in this property, not just the segment you're modifying. To make a modification, find the entire list of the current mappings being used to create the index by navigating to the URL

```
http://[HOST]:[ES_PORT]/liferay-[COMPANY_ID]/_mapping/LiferayDocumentType?pretty
```

Copy the contents in as the value of this property (either into System Settings or your OSGi configuration file). Leave the opening curly brace `{`, but delete lines 2-4 entirely:

```
"liferay-[COMPANY_ID]": {
    "mappings" : {
        "LiferayDocumentType" : {
```

Then, from the end of the mappings, delete the concluding three curly braces.

```
        }
    }
}
```

Now modify whatever mappings you'd like. The changes take effect once you save the changes and trigger a reindex from Server Administration. If you need to add new custom mappings without overriding any defaults, use `additionalTypeMappings` instead.

Here's a partial example, of a dynamic template that uses the analysis configuration above to analyze all string fields that end with `_ja`, overriding the default `template_ja` mapping.

```
{
    "LiferayDocumentType": {
        "dynamic_templates": [
            {
                "template_ja": {
                    "mapping": {
                        "analyzer": "kuromoji_liferay_custom",
                        "index": "analyzed",
                        "store": "true",
                        "term_vector": "with_positions_offsets",
                        "type": "string"
                    },
                    "match": "\\w+_ja\\b|\\w+_ja_[A-Z]{2}\\b",
                    "match_mapping_type": "string",
                    "match_pattern": "regex"
                }
                ...
            }
        ]
    }
}
```

Note: There's actually a third way to add configuration options to the Elasticsearch adapter. You or your favorite developer can publish a Settings Contributor Component and deploy it to Liferay's OSGi runtime. A developer tutorial on Search Extension Points will be written for this. In summary, the contributor module needs these things:

- A class that implements either `com.liferay.portal.search.elasticsearch.settings.SettingsContributor` or `com.liferay.portal.search.elasticsearch.settings.IndexSettingsContributor`.
  - If you're adding settings that would go into `additionalConfigurations`, override `SettingsContributor`.
  - If you want to add settings that would go into `additionalIndexConfigurations` or `additionalTypeMappings`, implement `IndexSettingsContributor`.
- An OSGi `@Component` annotation for either of the implementations mentioned in the last step.

Multi-line YAML Configurations

If you configure the settings from the last section using an OSGi configuration file, you might find yourself needing to write YAML snippets that span multiple lines. The syntax for that is straightforward and just requires appending each line with \n\, like this:

```
additionalConfigurations=\
                    cluster.routing.allocation.disk.threshold_enabled: false\n\
                    cluster.service.slow_task_logging_threshold: 600s\n\
                    index.indexing.slowlog.threshold.index.warn: 600s\n\
                    index.search.slowlog.threshold.fetch.warn: 600s\n\
                    index.search.slowlog.threshold.query.warn: 600s\n\
                    monitor.jvm.gc.old.warn: 600s\n\
                    monitor.jvm.gc.young.warn: 600s
```

Troubleshooting Elasticsearch

Sometimes things don't go as planned. If you've set up Liferay DXP with Elasticsearch in remote mode, but Liferay DXP can't connect to Elasticsearch, check these things:

- Cluster name: The value of the `cluster.name` property in Elasticsearch must match the `clusterName` property you configured for Liferay's Elasticsearch adapter.
- Transport address: The value of the `transportAddress` property in the Elasticsearch adapter must match the port where Elasticsearch is running. If Liferay DXP is running in embedded mode, and you start a standalone Elasticsearch node or cluster, it detects that port `9300` is taken and switches to port `9301`. If you then set Liferay's Elasticsearch adapter to remote mode, it continues to look for Elasticsearch at the default port (`9300`).

Now you have Elasticsearch configured for use. If you're a Liferay DXP customer, you can read here to learn about configuring Shield to secure your Elasticsearch data.

Elasticsearch Connector System Settings, By Operation Mode

Some of the settings available for the Elasticsearch connector are applicable for only one operation mode (REMOTE or EMBEDDED). Refer to the table below:

Adapter Setting/Operation Mode | EMBEDDED | REMOTE | `clusterName` | x | x `operationMode` | x | x `indexNamePrefix` | x | x `indexNumberOfReplicas*` | x | x `indexNumberOfShards*` | x | x `bootstrapMlockAll` | x | - `logExceptionsOnly` | x | x `retryOnConflict` | x | x `discoveryZenPingUnicastHostsPort` | x | - `networkHost` | x | - `networkBindHost` | x | - `networkPublishHost` | x | - `transportTcpPort` | x | - `transportAddresses` | - | x `clientTransportSniff` | - | x `clientTransportIgnoreClusterName` | - | x `clientTransportPingTimeout*` | - | x `clientTransportNodesSamplerInterval` | - | x `httpEnabled` | x | - `httpCORSEnabled` | x | - `httpCORSAllowOrigin`

| x | - `httpCORSConfigurations` | x | - `additionalConfigurations` | x | x `additionalIndexConfigurations` | x | x `additionalTypeMappings` | x | x `overrideTypeMappings` | x | x

* **Note:** Available in the Liferay Connector to Elasticsearch 6 only.

Related Topics

Introduction to Liferay Search

Customizing Liferay Search

1 This is, of course, a nod to all those fans of Boaty Mcboatface.

52.3 Securing Elasticsearch with Shield

Elasticsearch makes storing, searching, and analyzing your Liferay DXP search data easy. When it comes to securing that data, use Elasticsearch's Shield plugin. To use Shield with Liferay DXP, you need the *Enterprise Search-Standard* subscription. This subscription gives you access to an adapter plugin for configuring Liferay DXP for Shield.

With Shield you can prevent unauthorized users from accessing the Elasticsearch cluster, preserve data integrity, and create an audit trail to inspect suspicious activity. This guide shows you the basics of how to install and configure Shield, and then how to configure Liferay DXP for Shield, using a convenient Shield adapter plugin.

Note: The Shield plugin can only be used when you're running Elasticsearch in *remote mode*. If you're not sure what that means refer to the Configuring Elasticsearch article. It's not possible to install Shield into Liferay DXP's default embedded Elasticsearch—and you shouldn't be using embedded Elasticsearch in production anyway.

Here's the process for configuring Shield:

- Install the Shield plugin on Elasticsearch.
- Create a user for Liferay DXP, with user name and password.
- Install your Shield license.
- Install Liferay DXP's Shield adapter plugin and configure it.
- Enable Transport Layer Security (TLS) to encrypt your connection between Liferay DXP and Elasticsearch.

These terms will be useful to understand as you read this guide:

- *Elasticsearch Home* refers to the root folder of your unzipped Elasticsearch installation (for example, `elasticsearch-2.4.0`).
- *Liferay Home* refers to the root folder of your Liferay DXP installation. It contains the `osgi`, `deploy`, `data`, and `license` folders.

Installing Shield on Elasticsearch

First install the Shield plugin on your Elasticsearch cluster.

1. Navigate to Elasticsearch Home and install the license plugin and the Shield plugin by executing.

```
./bin/plugin install license
```

and then

```
./bin/plugin install shield
```

2. Next, you need to prepare for Shield to authenticate requests.

   Users making requests to an Elasticsearch installation protected by Shield must be part of the *realm*, a user database configured for Shield. You can use the native user management system built into Shield, called *esusers*, or you can use an external system like LDAP.

   Roles for Shield are defined in `[Elasticsearch_Home]/config/shield/roles.yml` and include these:

   `admin`: Has permission to perform any cluster or index action.

   `power_user`: Has permission to monitor the cluster and perform any index action.

   `user`: Has permission to perform read actions on any index.

   So who is the user you need to configure for Liferay? It's Liferay itself, and it needs the `admin` role. Liferay DXP's Elasticsearch client sends its authentication token (in other words, its user name and password) to Shield. Since Shield also has the authentication token stored in its user database, Liferay is a recognized user and has no problems communicating with the Elasticsearch cluster.

   From Elasticsearch Home, add an `admin` user named *liferay* to the esusers database:

   ```
   ./bin/shield/esusers useradd liferay -r admin
   ```

   When prompted, enter the password *liferay*.

   Note: Of course you can change these values if desired. Consider naming the user *shieldy_mcshieldface*, for example.1

3. Start Elasticsearch.

   ```
   ./bin/elasticsearch
   ```

4. Install your license file:

   ```
   curl -XPUT -u liferay 'http://localhost:9200/license?acknowledge=true' -d @license.json
   ```

   Enter the password you configured, and you'll get a confirmation message that looks like this:

   ```
   {"acknowledged":true}
   ```

5. To test that you have access to Elasticsearch, enter

   ```
   curl -u liferay:liferay -XGET 'http://localhost:9200/'
   ```

   and you'll see the Elasticsearch node information printed in the console:

```
{
  "name" : "Amphibius",
  "cluster_name" : "LiferayElasticsearchCluster",
  "version" : {
    "number" : "2.4.0",
    "build_hash" : "8ff36d139e16f8720f2947ef62c8167a888992fe",
    "build_timestamp" : "2016-01-27T13:32:39Z",
    "build_snapshot" : false,
    "lucene_version" : "5.5.2"
  },
  "tagline" : "You Know, for Search"
```

For more information on installing Shield, see the Elasticsearch documentation.

Once Shield is installed, you can configure Liferay DXP's Shield adapter.

Installing and Configuring Liferay DXP's Shield Adapter

On the Liferay DXP side of the equation, you need to configure the authentication token for the *liferay* Shield user you created in the previous section. Liferay DXP has a Shield adapter plugin for this purpose.

First install the Shield adapter plugin (called *7.0 Search Elasticsearch Shield*). Once the plugin is installed, there's a new Shield Configuration entry in the System Settings application (*Control Panel → Configuration → System Settings*), under the Foundation heading. Configure it so that its user name and password match the *liferay* user you added to Shield.

You can configure the Shield adapter in the System Settings section in the Control panel, or through an OSGi configuration file.

Follow these steps to configure the Shield adapter using an OSGi configuration file:

1. Create a file named `com.liferay.portal.search.elasticsearch.shield.configuration.ShieldConfiguration.cfg` in `[Liferay_Home]/osgi/configs`.

2. Add this content:

   ```
   password=liferay
   requiresAuthentication=true
   requiresSSL=false
   username=liferay
   ```

   For a description of all the Shield adapter's configuration options, see the section *Available Shield Adapter Configurations*.

3. Start Liferay DXP.

Follow these steps to configure the Shield adapter using System Settings:

1. Navigate to *Control Panel → Configuration → System Settings*, and click on the *Foundation* heading. Navigate to, or search for, the *Shield Configuration* entry and click on it.

2. Set the password to *liferay*, check the *Requires authentication* box, uncheck the *Requires SSL* box, and make sure the user name is *liferay*. Click *Update* when you're done.

   Note: If you set a different user name and password while configuring Shield, make sure those match the user name and password you configure here.

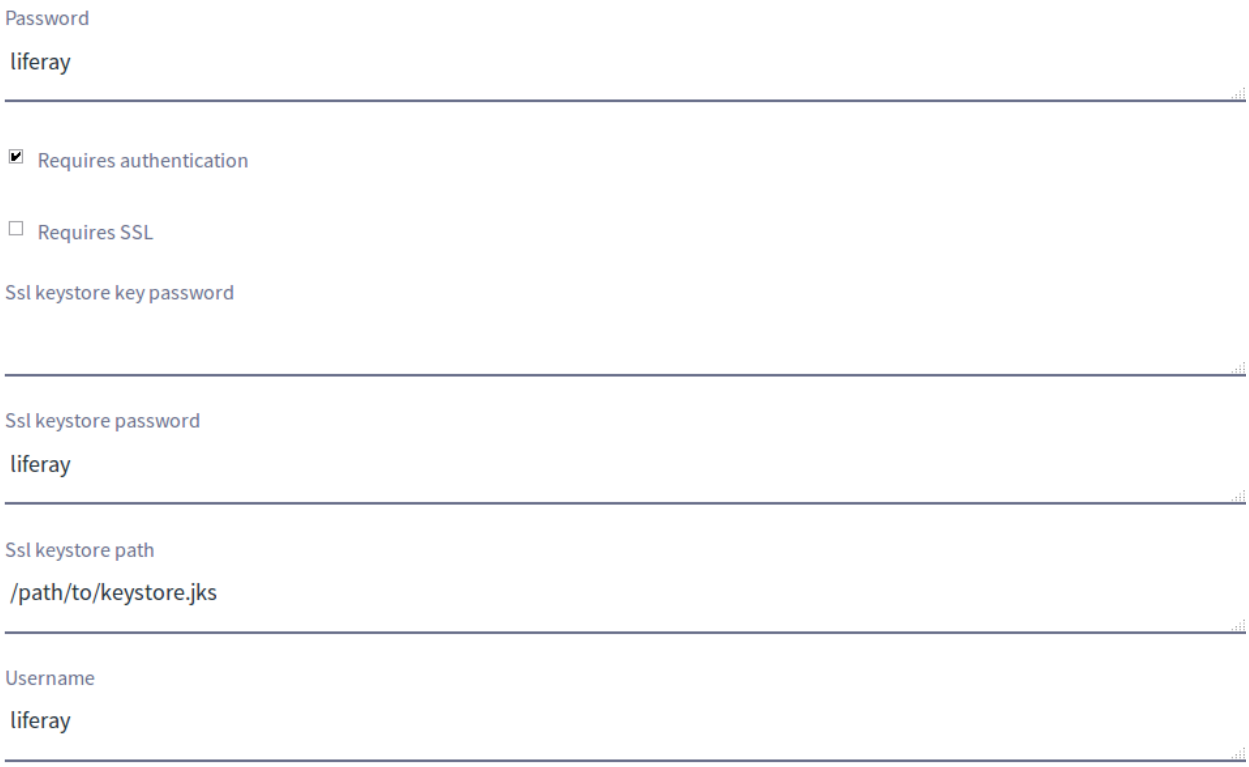


Figure 52.4: You can configure the 7.0 Search Elasticsearch Shield plugin from System Settings.

3. Now you can re-index against Elasticsearch, and your data is secured by Shield. To re-index, go to the Control Panel's Configuration section and click *Server Administration*. Find the Index Actions heading and click *Execute* next to *Reindex all search indexes.*

For a complete list of the Shield adapter's available configuration options, see here.

Encrypting Elasticsearch Connections

Your Elasticsearch connection now uses Shield to require authentication, but the authentication token is sent in plain text. For additional security, enable Transport Layer Security (TLS) encryption.

These instructions set up a *wildcard* certificate to be used across the entire cluster. See the Elasticsearch documentation for alternative configuration approaches and more information.

Note that for Elasticsearch to access your keystore, it must be placed under the `config` directory. Run the following commands under `Elasticsearch_Home/config` to configure SSL with an Elasticsearch cluster running on `localhost`, for example.

1. Stop Liferay DXP and Elasticsearch.

2. Set up a Certificate Authority (CA) for Shield. Refer to Elastic's article on Setting Up a Certificate Authority for the details.

   Note for Windows: In step 2 of the linked documentation on setting up a certificate, ensure that the `serial` file contains *01* with no quotation marks. Otherwise you'll encounter errors when you follow the steps below on signing the CSR.

3. Use the Java `keytool` command to create a new Java Keystore, import the CA that will issue the wildcard certificate:

```
keytool -importcert -keystore es-ssl.keystore.jks -file certs/cacert.pem -trustcacerts -storepass liferay -alias ca_cert
```

4. Create a private key in the Java Keystore:

```
keytool -storepass liferay -genkey -alias es-shield -keystore es-ssl.keystore.jks -keyalg RSA -keysize 2048 -validity 3650 -dname "cn=localhost"
```

5. Create a certificate signing request (CSR) for requesting a certificate from the issuing CA:

```
keytool -storepass liferay -certreq -alias es-shield -keystore es-ssl.keystore.jks -keyalg RSA -keysize 2048 -validity 3650 -dname "cn=localhost" > es-ssl.keystore.csr
```

6. Sign the CSR using Elastic's guide.

7. Once the CA has signed the CSR and returned the certificate in PEM format, import it into the Java Keystore:

```
keytool -storepass liferay -importcert -keystore es-ssl.keystore.jks -alias es-shield -file certs/01.pem
```

8. Add the following lines to `[Elasticsearch_Home]/config/elasticsearch.yml`:

```
shield.ssl.keystore.path: /path/to/es-ssl.keystore.jks
shield.ssl.keystore.password: liferay
shield.ssl.keystore.key_password: liferay
shield.transport.ssl: true
shield.http.ssl: true
```

   Here you're configuring Shield's SSL properties, including pointing to the keystore file you just generated. For more information on these settings, read here and here.

9. Update the Shield adapter configuration file you created earlier in `Liferay_Home/osgi/configs` by adding these lines:

```
requiresSSL=true
sslKeystorePath=/path/to/es-ssl.keystore.jks
sslKeystorePassword=liferay
```

   Now, in addition to enabling authentication, you're enabling SSL encryption and pointing Liferay DXP at the keystore file you created for Shield.

   Alternatively, you can configure these settings in System Settings. This will be more useful during development and testing.

10. Start Elasticsearch and Liferay DXP.

Now Shield is fully configured, with both authentication and encryption protecting your Elasticsearch cluster. Next, you can learn how to install and configure Marvel, Elasticsearch's monitoring plugin, to visualize the health and performance of your Elasticsearch cluster.

1 This is, of course, a nod to all those fans of [Boaty Mcboatface]

52.4 Tuning and Scaling Elasticsearch

Since search engines benefit heavily from caching, their JVM memory profiles are substantially different from those of a JVM focused on serving content and web views (e.g., a JVM running Liferay DXP). In production environments, search engines and Liferay DXP should therefore always be on separate JVMs.

The following sections provide a synopsis of Elasticsearch configurations. Prior to deployment, we strongly recommend reading Elastic's documentation on production deployment.

You'll learn how to configure these settings:

- JVM
- File System
- Scale

Read on to get Elasticsearch humming with Liferay DXP!

JVM

In general, you should allocate 45% of the available system memory to Elasticsearch, up to a maximum of 31 GB. You should configure heap sizing by setting the `ES_HEAP_SIZE` environment variable. Also, the JVM vendor and version for the Elasticsearch server must be identical to those of the Liferay DXP server.

File System

You should configure your OS for at least 64,000 file descriptors (the default Linux value is 1024). Elasticsearch also uses NioFS and MMapFS. You must therefore ensure there is sufficient virtual memory available for memory-mapped files. Consult your system administrator for information on how to configure these values.

Scaling Your Deployment

Proper scaling and tuning of an Elasticsearch cluster primarily depends on the type of indexes it holds and how they're intended to be used. Since Liferay DXP is a flexible development platform, no two applications index and search for data in exactly the same way. Read the definitive Elasticsearch guide, and understand the differences between indexing-intensive applications and search-intensive applications. Then you'll be able to predict usage patterns for your Liferay DXP indexes and design the optimally scaled and tuned cluster.

Always perform tests to ensure optimal configurations.

52.5 Installing Marvel for Elasticsearch

Marvel is a monitoring tool for Elasticsearch. It lets you view the performance and health of your Elasticsearch cluster, so you can anticipate issues ahead of time, troubleshoot them quickly, and scale your cluster appropriately.

To obtain Marvel and the client application for Liferay DXP, you need the *Enterprise Search-Standard* subscription. With it you'll get a connector plugin for configuring Liferay DXP for Marvel and a Marvel Portlet that can be added to a page. The Marvel portlet gives you access to all of Marvel's functionality. If you don't yet have an Enterprise Search subscription, contact your sales representative.

This article shows you how to install and configure Marvel for Liferay DXP–and Liferay DXP for Marvel–with these general steps:

- Install Marvel and Kibana on Elasticsearch.

```
**Note:** If you're wondering what
[Kibana](https://www.elastic.co/products/kibana) is, it's the visualization
piece of the equation. Elasticsearch is the search engine, and a Marvel
agent in Elasticsearch collects and sends data from Elasticsearch to
Kibana. Kibana, including a Marvel UI plugin, displays the Marvel agent's
data.
```

- Configure Kibana to work with Shield and to be accessed through the Marvel Portlet.
- Configure Liferay DXP's Marvel adapter plugin.
- Add the Marvel portlet to a page and start monitoring your cluster.

Note: This tutorial shows you how to get Marvel up and running by installing it onto your Elasticsearch cluster directly. The best approach is to follow Elastic's guide and set up the Marvel cluster separately from the production cluster, so that a troublesome Elasticsearch cluster does not inhibit your ability to use Marvel to diagnose its problems. In short, it involves these steps:

1. Install the Marvel agent and License plugins on the production Elasticsearch cluster.
2. Install a separate Elasticsearch cluster (the monitoring cluster).
3. Configure your production cluster's nodes to send Marvel data to the monitoring cluster.
4. Download and install Kibana on the monitoring cluster's machine.
5. Install the Marvel app into Kibana.
6. Configure Kibana to connect to the monitoring cluster.
7. Live long and prosper.

These terms will be useful to understand as you read this guide:

- *Elasticsearch Home* refers to the root folder of your unzipped Elasticsearch installation (for example, `elasticsearch-2.4.0`).
- *Liferay Home* refers to the root folder of your Liferay DXP installation. It contains the `osgi`, `deploy`, `data`, and `license` folders.
- *Kibana Home* refers to the root folder of your Kibana installation.

Installing Kibana and Marvel

Before you install Kibana or Marvel, make sure you've read and followed the instructions on installing and configuring Elasticsearch for Liferay DXP.

1. Install the `marvel-agent` plugin on Elasticsearch by navigating to Elasticsearch Home and entering

   ```
   ./bin/plugin install marvel-agent
   ```

2. Download a compatible version of Kibana and extract it to your Liferay Home folder.

   Note: Your Liferay Home folder should now have the usual Liferay DXP folders, as well as the `elasticsearch-[version]` (Elasticsearch Home) and `kibana-[version]` (Kibana Home) folders. A Liferay Home folder for a Tomcat bundle would look like this:

```
Liferay_Home/
    data/
    deploy/
    Elasticsearch_Home/
    Kibana_Home/
    license/
    logs/
    osgi/
    patching-tool
    tomcat-[version]
```

3. Install Marvel on Kibana by navigating to Kibana Home and entering

```
./bin/kibana plugin --install elasticsearch/marvel/[version]
```

The next step is to configure Kibana to connect with Elasticsearch. The instructions vary depending on whether you are using Shield or not.

Configuring Kibana for Elasticsearch

Now you need to configure Kibana. Since you'll use Liferay DXP's Marvel Portlet as a proxy servlet to view the Kibana UI and Marvel, you'll start configuring that here.

If you're using Shield, follow all the steps below. If not, skip the steps that begin with *[Shield]*.

1. *[Shield]* Set the user name and password in `Kibana_Home/config/kibana.yml`, by entering these lines:

```
elasticsearch.username: liferay
elasticsearch.password: liferay
elasticsearch.url: "https://<your_elasticsearch_host>:9200"
```

2. *[Shield]* Also in `kibana.yml`, you need to add Elasticsearch's Certificate Authority (CA) property. If you're using a self-signed certificate as demonstrated in the Shield article, export the certificate from the JKS file and use it as the CA.

   From `Elasticsearch_Home/config/path-to-your-JKS`, execute

```
keytool -v -importkeystore -srckeystore es-ssl.keystore.jks -srcalias es-shield -destkeystore es-ssl.PKCS12.p12 -deststoretype PKCS12
```

   This command converts the JKS file to a PKCS12 file, which is more portable.

   Note: You'll be prompted to enter a password for the new PKCS12 file. Enter *liferay*. You'll also be prompted for the password of the JKS file, which is also *liferay* if you followed the instructions from the Shield article.

3. *[Shield]* From `Elasticsearch_Home/config/path-to-your-PKCS12` enter

```
 openssl pkcs12 -in es-ssl.PKCS12.p12 -out es-ssl.CA.pem
```

   to export the certificate from the PKCS12 file. When prompted for the password to access the PKCS12 file, enter *liferay*. Likewise, enter *liferay* when prompted to create a password for the PEM file.

4. *[Shield]* Move the PEM file created above to `Kibana_Home/config`.

5. *[Shield]* Add the following line to `Kibana_Home/config/kibana.yml`:

```
elasticsearch.ssl.ca: /path/to/Kibana_Home/config/es-ssl.CA.pem
```

6. Configure Kibana to be accessed through the Liferay DXP Marvel Portlet by adding the following to `kibana.yml`:

   ```
   server.basePath: "/o/portal-search-elasticsearch-marvel-web/marvel-proxy"
   ```

   Note: the `/o` prefix is a default. Change it if you're running Liferay DXP under a different web context path.

7. Start Elasticsearch by running

   ```
   ./bin/elasticsearch
   ```

   from Elasticsearch Home.

8. Start Kibana by running

   ```
   ./bin/kibana
   ```

   from Kibana Home.

9. Start Liferay DXP.

Configuring SSL on Kibana

To run Kibana with SSL encryption you'll need to do these things:

- Place a proxy server in front of it.
- Tell the proxy's firewall to allow port 5601 (Kibana's port).
- If using a self signed certificate in the proxy server, `ProxyServlet` won't be able to trust the connection, so add the certificate to the JVM truststore.

Note: Liferay DXP does not support protecting your Kibana connection with Shield. If you try to secure Kibana with Shield, Marvel will not work inside Liferay DXP's portlet.

Configuring Liferay's Marvel Adapter

Now that you have Marvel and Kibana configured, you can configure the Marvel adapter in Liferay DXP.

There's a *Marvel* entry in the System Settings application (*Control Panel → Configuration → System Settings*) under the Foundation heading.

There are several configuration options for the Marvel adapter plugin. While you can make edits directly in System Settings, in production environments you'll want to make edits to the defaults through a `.cfg` configuration file. Name it `com.liferay.portal.search.elasticsearch.marvel.web.configuration.MarvelWebConfiguration.cfg` and place it in `Liferay_Home/osgi/configs`.

Here are the settings you can configure:

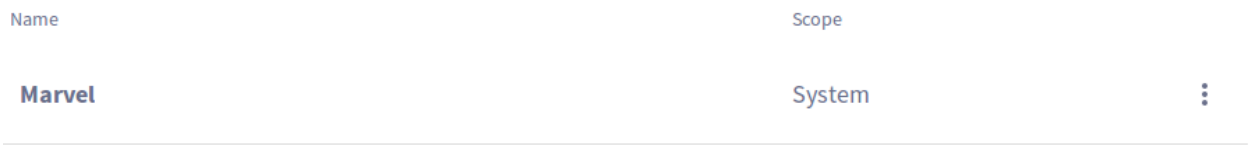


Figure 52.5: The Marvel adapter in Liferay DXP can be configured directly from System Settings. This is most useful during testing and development.

`kibanaURL=http://localhost:5601` Set the String URL for the remote Kibana server where Marvel is deployed. The URL should only contain the host and port. If you're using a Proxy Server in front of Kibana, set the Proxy Server's URL here instead of Kibana's URL.

`proxyServletLogEnable=false` Set this boolean to true to show debug logging for the Marvel proxy. This is useful for troubleshooting URL mappings passing through the proxy from Liferay DXP to Kibana, and to display the error codes from requests that Kibana denies.

`shieldUserName=liferay` If Shield is being used, set the String value for the user name used for authenticating to Shield. This setting is ignored if Shield isn't installed.

`shieldPassword=liferay` Set the String password used for Shield authentication. This setting is ignored if Shield isn't configured.

Accessing Marvel's UI in Liferay

As mentioned earlier, Liferay provides a Marvel Portlet that displays an embedded, proxied version of the Marvel UI from Kibana.

If you've followed the steps outlined above, and you have Elasticsearch, Marvel, Kibana, and Liferay DXP configured correctly, you can just add the Marvel Portlet to a page in Liferay DXP:

1. Click the page's *Add* button ( ) and select *Applications*.
2. Enter *Marvel* in the search box, and click *Add* next to the Marvel entry in the Tools category. First you'll see a message about Kibana loading, then the Marvel UI appears.
3. You're ready to start exploring Marvel's monitoring interface. For an overview of your cluster, click on its name in the *Your Clusters* table (if you left it at the default, it's `LiferayElasticsearchcluster`).

For more information on what Marvel offers you, refer to Elasticsearch's Marvel guide.

With Liferay DXP's *Enterprise Search-Standard* subscription, you not only have a powerful search engine, but you have security and monitoring tools at your disposal. You can now diagnose, troubleshoot, and fix problems more easily than ever.

52.6 Using Solr

Solr is a popular enterprise search platform build on Apache Lucene. It's popular for its reliability, scalability, and fault tolerance. Read more about it here.

Although Elasticsearch is the default search engine that ships with Liferay DXP, it's perfectly valid to use Solr instead. In particular, if you've already been using Solr with a previous version of Liferay DXP, or your deployment system (for example, your OS or JVM) isn't supported by Elasticsearch, you might choose to use Solr to search and index your Liferay DXP data.

Liferay's support for Solr is compatible with Solr versions 5.2.x through 5.5.x. To make Liferay DXP and Solr talk to each other, you'll need to install the Liferay Solr adapter. There are two ways to do this:

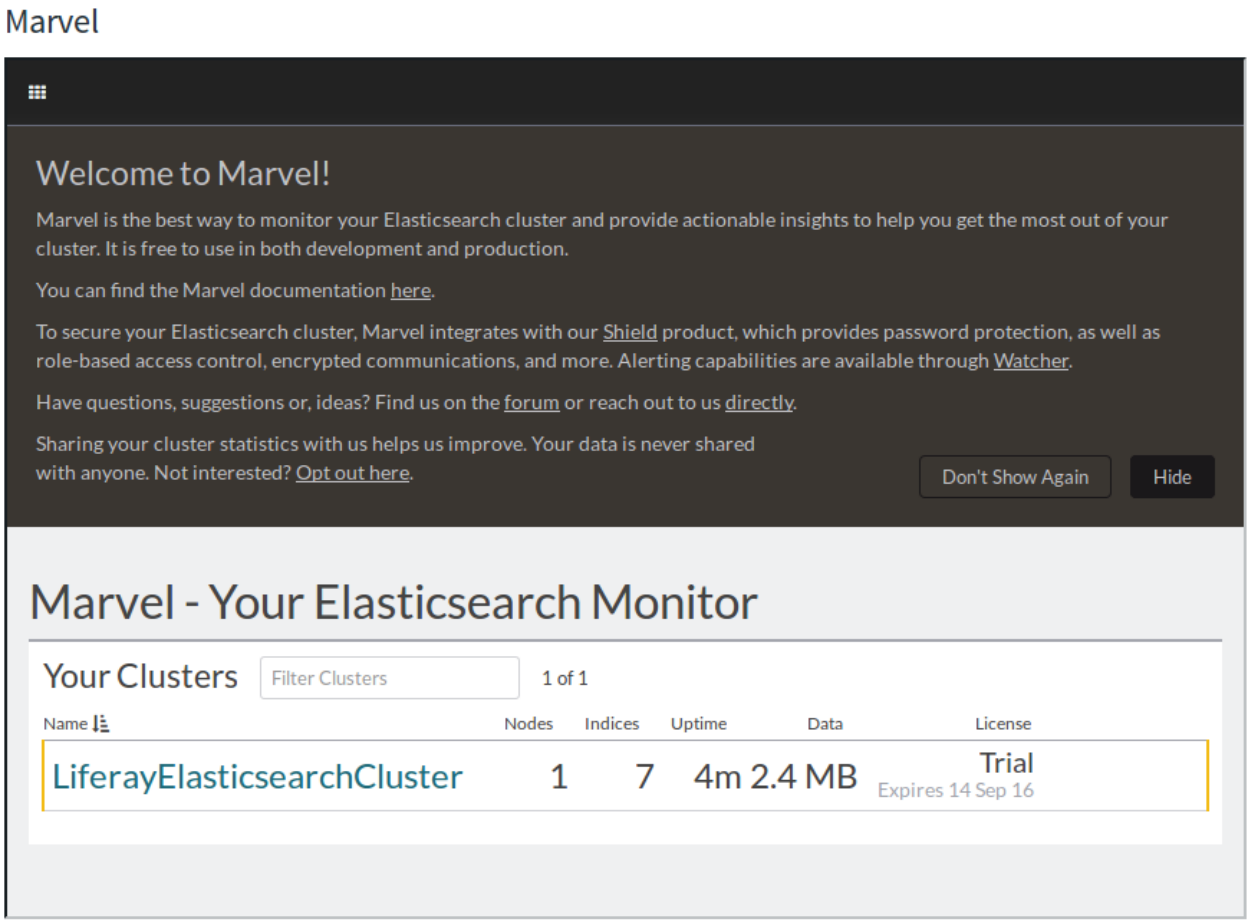


Figure 52.6: You can monitor your Elasticsearch cluster from Liferay DXP using the Marvel Portlet.

1. Navigate to the Liferay Marketplace website and download the LPKG file for Liferay (CE) Solr 5 Search Engine. Once you do, copy the LPKG to your `Liferay_Home/osgi/marketplace` folder.

2. In your running Liferay instance, navigate to *Control Panel* → *Apps* → *Store*. Sign in using your credentials, search for Solr Search Engine, and purchase (it's free) the Liferay (CE) Solr 5 Search Engine entry.

This guide leads you through the process of installing and configuring Solr. As you proceed, these terms will be useful to keep in mind:

Solr Home: The center of the Solr system (pun intended). This directory is `solr-[version]/server/solr`.

Liferay Home: The root folder of your Liferay DXP installation. It contains the `osgi`, `deploy`, `data`, and `license` folders, among others.

Before configuring Liferay DXP for Solr, you need to install and set up Solr.

Installing and Configuring Solr 5

To install and properly configure Solr for Liferay DXP:

1. Download Solr and unzip it.

2. Navigate to `solr-[version]/server/solr`. This is Solr Home.

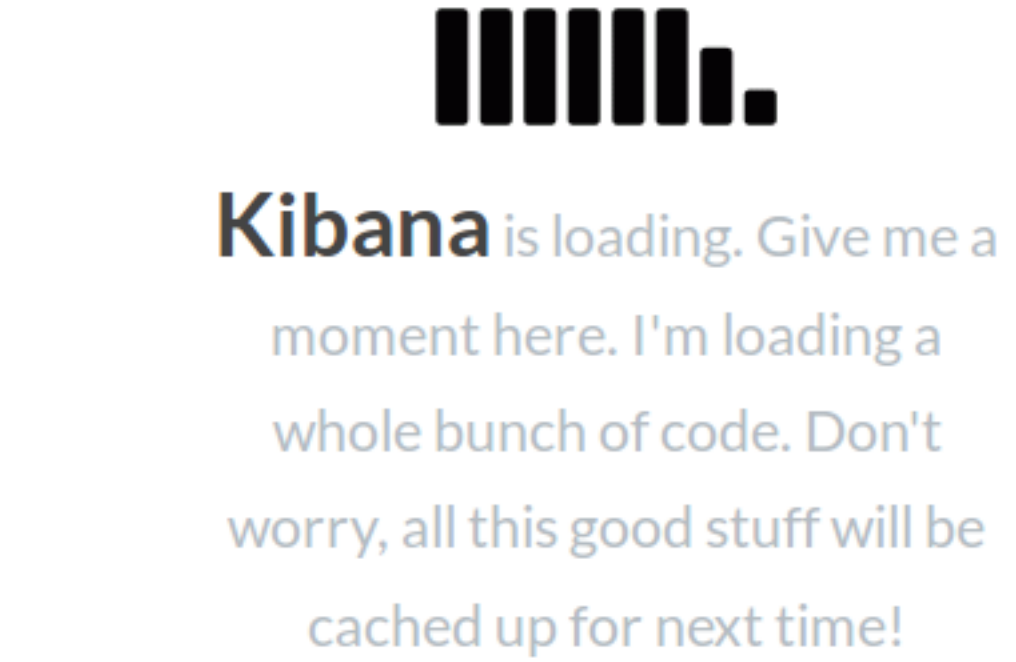


Figure 52.7: When you first add the Marvel Portlet to the page, Kibana fetches the Marvel content in the Marvel Portlet, and then your Marvel UI appears

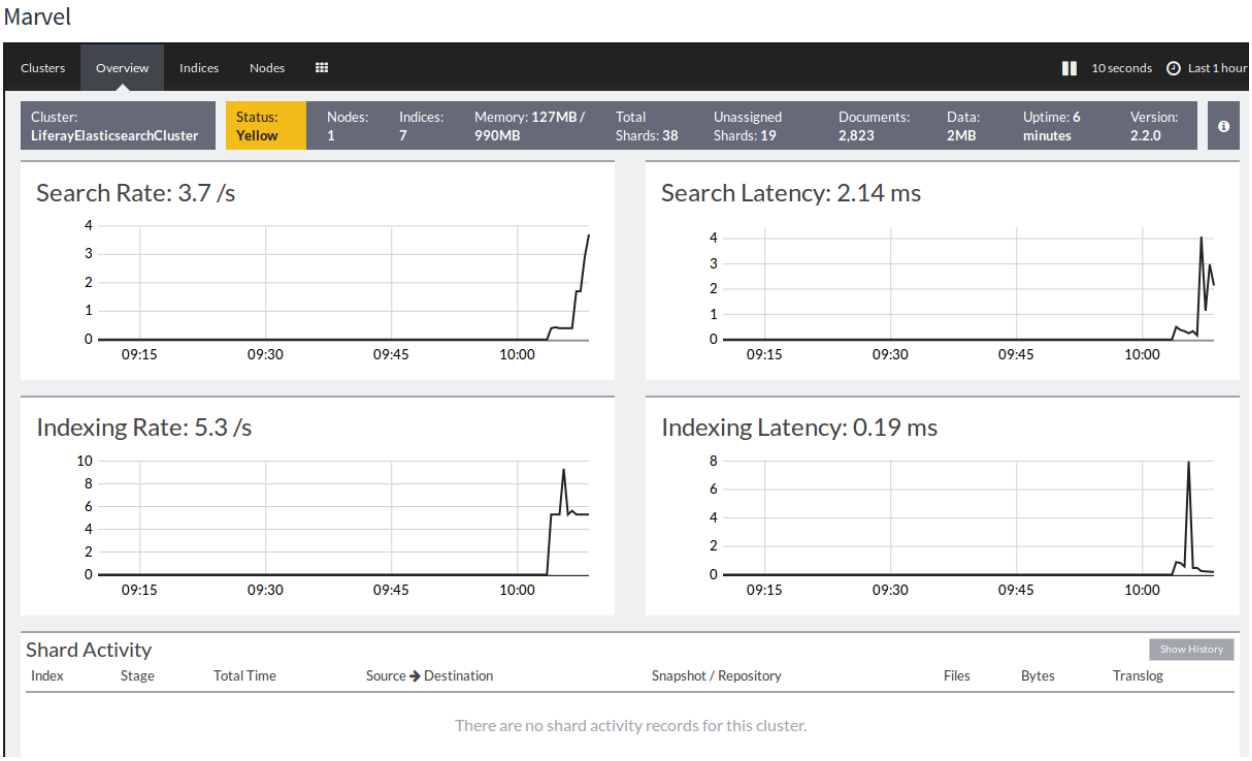


Figure 52.8: You can monitor the health of your cluster using the Marvel Portlet.

3. Create a new folder called `liferay`.
4. In the `liferay` folder, create two new folders: `conf` and `data`.
5. Copy the contents of `Solr_Home/configsets/data_driven_schema_configs/conf` to `Solr_Home/liferay/conf`.
6. Open the Liferay Solr Adapter's LPKG file with an archive manager.
   Open the `com.liferay.portal.search.solr.jar` file, and extract

   ```
   META-INF/resources/solrconfig.xml
   ```

   and

   ```
   META-INF/resources/schema.xml
   ```

   to

   ```
   Solr_Home/liferay/conf
   ```

   This replaces the current `solrconfig.xml` and `schema.xml` files with ones that tell Solr how to index data coming from Liferay DXP.
7. Create a `core.properties` file in `Solr_Home/liferay`, and add these contents:

   ```
   config=solrconfig.xml
   dataDir=data
   name=liferay
   schema=schema.xml
   ```

8. Checkpoint: your `Solr_Home/liferay` folder should now have this structure:

   ```
   liferay
   ├── conf
   │   ├── currency.xml
   │   ├── elevate.xml
   │   ├── lang
   │   ├── managed-schema
   │   ├── params.json
   │   ├── protwords.txt
   │   ├── schema.xml
   │   ├── solrconfig.xml
   │   ├── stopwords.txt
   │   └── synonyms.txt
   ├── core.properties
   └── data
   ```

9. Start the Solr server by entering

   ```
   ./bin/solr start -f
   ```

   from the top-level folder of your Solr installation (`solr-[version]`).
10. The Solr server listens on port `8983` by default. Navigate to `http://localhost:8983/solr/#/~cores` (assuming you're testing locally with `localhost` as your host), and confirm that the `liferay` core is available.

Solr is now installed. Next install and configure Liferay DXP's Solr adapter.

Installing and Configuring the Liferay Solr Adapter

Since Elasticsearch is the default search engine in Liferay DXP, the Elasticsearch adapter is already installed and running. Stop it before configuring the Solr adapter.

Stop the Elasticsearch adapter bundle using the App Manager, the Felix Gogo shell, or the bundle blacklist. If you're a Digital Enterprise customer, use the blacklist feature as described below. The App Manager and Gogo shell rely on the `osgi/state` folder to "remember" the state of the bundle. If you delete this folder (recommended during patching) the Elasticsearch connector will be reinstalled and started automatically.

Navigate to Control Panel → Apps → App Manager.

Once you're in the App Manager, search for *elasticsearch*. Find the Liferay Portal Search Elasticsearch module and click the edit (()) button. Choose the Deactivate option. This leaves the bundle installed, but stops it in the OSGi runtime.

Alternatively, use the Felix Gogo shell to stop the Elasticsearch adapter. First, open a Gogo shell and enter

```
lb elasticsearch
```

You'll see an active bundle named Liferay Portal Search Elasticsearch (version) listed in the Gogo shell.

```
ID |State      |Level|Name
239|Active     |   10|Liferay Portal Search Elasticsearch (2.0.4)
```

Stop the Elasticsearch adapter by entering

```
stop [bundle ID]
```

In the case above, the `[bundle ID]` is `239`.

Liferay Digital Enterprise: Digital Enterprise customers should blacklist the Elasticsearch, Shield, and Marvel plugins.

1. Create a

   ```
   com.liferay.portal.bundle.blacklist.internal.BundleBlacklistConfiguration.config
   ```

   file with these contents:

   ```
   blacklistBundleSymbolicNames=["com.liferay.portal.search.elasticsearch","com.liferay.portal.search.elasticsearch.shield","com.liferay.portal.search.
   ```

2. Place the file in `Liferay Home/osgi/configs`.

Now you can install and configure the Solr adapter:

1. Start Liferay DXP, then deploy the Solr adapter by copying the LPKG you downloaded to `Liferay_Home/deploy`.

   You'll see a `STARTED` message in your Liferay DXP log once the solr adapter is installed. Here's what the log message looks like:

   ```
   08:48:24,165 INFO [localhost-startStop-1][BundleStartStopLogger:35] STARTED
   com.liferay.portal.search.solr_2.0.3 [47]
   ```

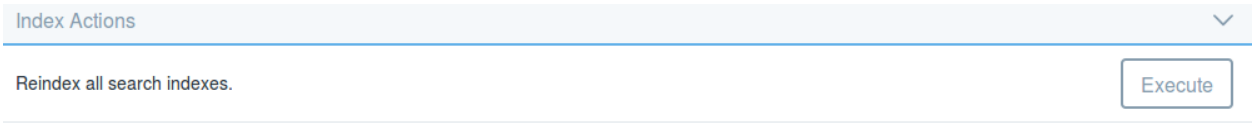


Figure 52.9: Once the Solr adapter is installed, you can reindex your Liferay DXP data against your Solr server.

2. To reindex against Solr, navigate to *Control Panel* → *Configuration* → *Server Administration*, and click *Execute* next to the *Reindex all search indexes* option.

In production deployments, specify your edits to the Solr adapter's default configurations using a configuration file deployed to the `Liferay_Home/osgi/configs` folder. Name the file

```
com.liferay.portal.search.solr.configuration.SolrConfiguration.config
```

During testing and development, the System Settings app in Liferay DXP's Control Panel → Configuration section is convenient for editing the default configurations.

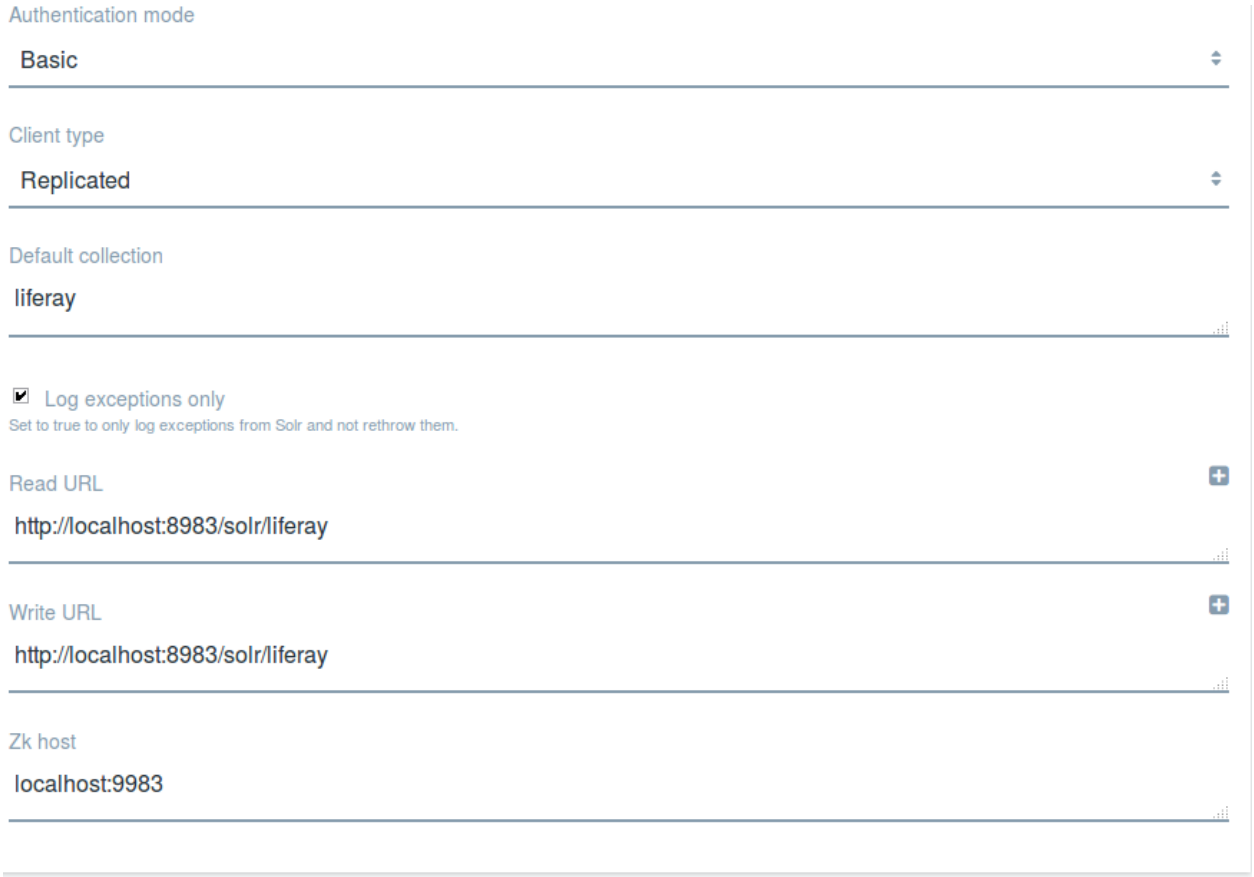


Figure 52.10: You can configure Solr from Liferay DXP's System Settings application. This is most useful during development and testing.

High Availability with SolrCloud

You can use SolrCloud if you need a cluster of Solr servers featuring fault tolerance and high availability. Note that to use SolrCloud in production, you should set up an external ZooKeeper ensemble. ZooKeeper is a centralized coordination service for managing distributed systems, such as your SolrCloud cluster.

The steps included here should be considered the bare minimum of what must be done to configure SolrCloud with Liferay DXP. For example, these instructions cover configuring SolrCloud on a single machine, whereas a production environment would feature multiple physical or virtual machines. These instructions also assume you've followed the earlier section on *Installing and Configuring Solr 5*. Refer to the SolrCloud guide for more information.

1. Stop the Solr server if it's running.
2. Navigate to the `Solr_Home/configsets` folder and create a folder called

   ```
   liferay_configs
   ```

3. Copy the `conf` folder from `Solr_Home/liferay` to the `liferay_configs` folder you just created.

   The `configset/liferay_configs` folder is used to configure the SolrCloud Liferay DXP collection, and is uploaded to ZooKeeper. By copying the `conf` folder from the `liferay` server configured earlier, you're using the `schema.xml` and `solrconfig.xml` files provided with the Liferay Solr Adapter.

4. Next launch an interactive SolrCloud session to configure your SolrCloud cluster. Use this command:

   ```
   ./bin/solr -e cloud
   ```

5. Complete the setup wizard. These steps demonstrate creating a two-node cluster:

   - Enter `2` for the number of nodes.
   - Specify ports `8983` and `7574` (the defaults). Both nodes are started with the start commands printed in the log:

     ```
     Starting up SolrCloud node1 on port 18983 using command:

     solr start -cloud -s example/cloud/node1/solr -p [port#]  -m 512m
     ```

   - Name the collection *liferay*.
   - Split the collection into two shards.
   - Specify two replicas per shard.
   - When prompted to choose a configuration, enter *liferay_configs*. You should see a log message that concludes like this when the cluster has been started:

     ```
     SolrCloud example running, please visit http://localhost:8983/solr
     ```

Now you have a new collection called *liferay* in your local SolrCloud cluster. Verify its status by running the *status* command:

```
./bin/solr status
```

You'll see log output like this:

```
Found 2 Solr nodes:

Solr process 12755 running on port 7574
{
  "solr_home":"/home/russell/Documents/docs-projects/solr-docs/solr-5.2.1/example/cloud/node2/solr/",
  "version":"5.2.1 1684708 - shalin - 2015-06-10 23:20:13",
  "startTime":"2016-08-19T18:11:27.087Z",
  "uptime":"0 days, 0 hours, 13 minutes, 2 seconds",
  "memory":"50 MB (%10.2) of 490.7 MB",
  "cloud":{
    "ZooKeeper":"localhost:9983",
    "liveNodes":"2",
    "collections":"1"}}


Solr process 12564 running on port 8983
{
  "solr_home":"/home/russell/Documents/docs-projects/solr-docs/solr-5.2.1/example/cloud/node1/solr/",
  "version":"5.2.1 1684708 - shalin - 2015-06-10 23:20:13",
  "startTime":"2016-08-19T18:11:21.637Z",
  "uptime":"0 days, 0 hours, 13 minutes, 8 seconds",
  "memory":"44.9 MB (%9.2) of 490.7 MB",
  "cloud":{
    "ZooKeeper":"localhost:9983",
    "liveNodes":"2",
    "collections":"1"}}
```

To stop Solr while running in SolrCloud mode, use the *stop* command, like this:

```
bin/solr stop -all
```

Configure the Solr Adapter for SolrCloud

There's only one thing left to do: specify the client type as *CLOUD* in Liferay's Solr adapter.

1. From System Settings or your OSGi configuration file, set the *Client Type* to *CLOUD*.

    ```
    clientType="CLOUD"
    ```

2. Start Liferay DXP if it's not running already.

Note: For a complete list of settings available in the Solr adapter, see the Solr Settings reference article.

Now you're able to configure Liferay DXP for Solr, and Solr for Liferay DXP. Remember that Elasticsearch is the default search engine for Liferay DXP, so if you're not constrained to use Solr or already a Solr expert, consider Elasticsearch for you search engine requirements. If you do use Solr, then you can tell all your colleagues that your Liferay DXP installation's search capability is Solr powered (pun intended).

CHAPTER 53

BACKING UP LIFERAY ENTERPRISE SEARCH

Liferay DXP's search solution is pluggable. Once you install a search engine and configure the appropriate adapter, you're using a standalone installation of the search engine (usually installed on a cluster of servers) to create and manage your search indices. That means there's really nothing special to do in the Liferay DXP installation when backing up your search engine's data. In large part, these articles summarize the process and point you to the relevant documentation on backing up Elasticsearch and Solr, the two search engines supported for plugging into Liferay.

53.1 Backing Up Elasticsearch

Elasticsearch replicas protect you against a node going down here or there, but they won't help you in the event of a catastrophic failure. Only good backup practices can help you in that case.

The process for backing up and restoring your Elasticsearch cluster takes three steps:

- Configure a repository
- Make a snapshot of the cluster
- Restore from the snapshot

For more detailed information on the process, refer to the Elasticsearch administration guide, and in particular to the documentation on the Snapshot/Restore module and on backing up your cluster.

Configuring a Repository

First configure a repository where your snapshots will be kept. Several repository types are supported:

- Shared file system, such as a Network File System or NAS
- Amazon S3
- HDFS (Hadoop Distributed File System)
- Azure Cloud

If using a shared file system repository type, first register the path to the shared file system in each node's `elasticsearch.yml` using the path.repo setting.

```
path.repo: ["path/to/shared/file/system/"]
```

When you create the repository using the Elasticsearch API, you can now refer to this repository location in your PUT command:

```
curl -XPUT localhost:9200/_snapshot/test_backup -d '{"type": "fs", "settings": { "location":"/path/to/shared/file/system/" } }'
```

In production replace `localhost:9200` with the proper `hostname:port` combination for your system, and replace `test_backup` with the name of the repository you want to create. Additionally, use the real path to your shared file system.

If the repository is successfully set up you'll see this message in your terminal:

```
{"acknowledged":true}
```

Once you have a repository, you can start creating snapshots.

Snapshotting the Cluster

The easiest approach is to create is a snapshot of all the indexes in your cluster. Here's the basic command for snapshotting everything:

```
curl -XPUT localhost:9200/_snapshot/test_backup/snapshot_1
```

If successful you see `{"accepted":true}` in the terminal.

You don't have to include all indexes in a snapshot. For example, if you're using Marvel, you might not want to include all the Marvel indexes. In this case just explicitly declare the indexes you want to include in the snapshot (leaving out the `.marvel` indexes):

```
curl -XPUT localhost:9200/_snapshot/test_backup/snapshot_2
{ "indices": "liferay-0,liferay-20116" }
```

It's important to note that Elasticsearch uses a *smart snapshotting* approach. To understand what that means, consider a single index. The first snapshot includes a copy of the entire index, while subsequent snapshots only include the delta between the first, complete index snapshot and the current state of the index.

Eventually you'll end up with a lot of snapshots in your repository, and no matter how cleverly you name the snapshots, you may forget what some snapshots contain. For this purpose, the Elasticsearch API includes the ability to get information about any snapshot. For example:

```
curl -XGET localhost:9200/_snapshot/test_backup/snapshot_1
```

returns

```
{"snapshots":[
    {"snapshot":"snapshot_1",
    "version_id":2020099,
    "version":"2.4.0",
    "indices":["liferay-0","liferay-20116"],
    "state":"SUCCESS",
    "start_time":"2016-11-29T19:50:12.375Z",
    "start_time_in_millis":1480449012375,
    "end_time":"2016-11-29T19:50:13.654Z",
    "end_time_in_millis":1480449013654,
    "duration_in_millis":1279,
    "failures":[],
    "shards":{
        "total":10,
        "failed":0,
        "successful":10
```

```
        }
    }
]}
```

There's lots of useful information here, including which indexes were included in the snapshot.

If you want to get rid of a snapshot, use the DELETE command.

```
curl -XDELETE localhost:9200/_snapshot/test_backup/snapshot_1
```

You might trigger creation of a snapshot and regret it (for example, you didn't want to include all the indexes in the snapshot). If you're snapshotting a lot of data, this can cost time and resources. To cancel the ongoing creation of a snapshot, use the same DELETE command. The snapshot process is terminated and the partial snapshot is deleted from the repository.

Restoring from a Snapshot

What good is a snapshot if you can't use it to restore your search indexes in case of catastrophic failure? Restoring your cluster from a snapshot is easy. You'll leverage the `_restore` API:

```
curl -XPOST localhost:9200/_snapshot/test_backup/snapshot_1/_restore
```

This command restores all the indexes in the snapshot. If you want to restore only specific indexes from a snapshot, you can. For example, enter

```
curl -XPOST
localhost:9200/_snapshot/test_backup/snapshot_1/_restore
{
    "indices": "liferay-20116",
    "rename_pattern": "liferayindex_(.+)",
    "rename_replacement": "restored_liferayindex_$1"
}
```

This restores only the index named `liferay-20116index_1` from the snapshot. The `rename...` settings specify that the beginning `liferayindex_` are replaced with `restore_liferayindex_`, so `liferay-20116index_1` becomes `restored_liferay-20116index_1`.

As with the snapshotting process, an errant restored index can be canceled with the DELETE command:

```
curl -XDELETE localhost:9200/restored_liferay-20116index_3
```

Nobody likes catastrophic failure on a production system, but Elasticsearch's API for snapshotting and restoring indexes can help you rest easy knowing that your search cluster can be restored if disaster strikes.

53.2 Upgrading to Elasticsearch 6

Elasticsearch 6 is supported for Digital Enterprise subscribers. The exact supported Elasticsearch version range depends on the Fix Pack Level: consult the compatibility matrix for details. Community Edition users running 7.0 CE GA 7 or greater can use up to Elasticsearch 6.1.x. If you're not already running a remote Elasticsearch 2.x server, follow the installation guide to install Elasticsearch 6 and the configuration guide to configure the Elasticsearch adapter. Here, you'll learn to upgrade an existing Elasticsearch 2.x server (or cluster) to Elasticsearch 6.5.x:

1. Install and configure Elasticsearch 6.5.x.

2. Download the Elasticsearch 6 adapter from Liferay Marketplace.
3. Stop the default Elasticsearch adapter.
4. Stop the Elasticsearch 2.4 server.
5. Start Elasticsearch 6.
6. Install and configure the Elasticsearch 6 adapter.
7. Re-index all search and spell check indexes.

Before Proceeding: Back up your existing data before upgrading Elasticsearch. If something goes wrong during or after the upgrade, roll back to 2.x using the uncorrupted index snapshots. See here for more information.

Key Changes: This list compiles known changes within Elasticsearch that are likely to affect Liferay DXP users. For a more complete reckoning of what's changed in Elasticsearch, see the breaking changes for Elasticsearch 6.5.

Custom Mappings

Aggregations: Aggregating on analyzed `String` fields was possible in Elasticsearch 2.x, but aggregating on analyzed `text` fields is not advised in Elasticsearch 6.5. Instead, use a `keyword` field or add field data. Beware; `fielddata` is disabled by default because it is memory-intensive.

Spell Check and Query Suggestions

Due to Elastic's removal of mapping types from search documents, spell check indexes and suggestion dictionaries now use a single document type. If your installation was leveraging either functionality, you must re-index all spell check indexes (from Control Panel → Configuration → Server Administration).

Installing Elasticsearch 6.5

1. Download Elasticsearch 6.5.x and unzip it wherever you please.
2. Name the cluster by configuring the `cluster.name` property in `elasticsearch.yml`:

```
cluster.name: LiferayElasticsearchCluster
```

3. Install the following required Elasticsearch plugins:

    - `analysis-icu`
    - `analysis-kuromoji`
    - `analysis-smartcn`
    - `analysis-stempel`

4. To install these plugins, navigate to Elasticsearch Home and enter

```
./bin/elasticsearch-plugin install [plugin-name]
```

5. Replace *[plugin-name]* with the Elasticsearch plugin's name.
6. Once installed, start Elasticsearch 6 by running

```
./bin/elasticsearch
```

from Elasticsearch Home.

Download the Elasticsearch 6 Adapter

Download the Elasticsearch 6 adapter LPKG from Liferay Marketplace.

Stop the Elasticsearch Adapter and Elasticsearch 2.x

Before installing the Elasticsearch 6 adapter, you must stop the running Elasticsearch adapter that ships with Liferay DXP.

Blacklist the Elasticsearch, Shield, and Marvel plugins. Create a

```
com.liferay.portal.bundle.blacklist.internal.BundleBlacklistConfiguration.config
```

file with these contents:

```
blacklistBundleSymbolicNames=["com.liferay.portal.search.elasticsearch","com.liferay.portal.search.elasticsearch.shield","com.liferay.portal.search.elastics
```

Place the file in `Liferay Home/osgi/configs`.

Note: If you're a Digital Enterprise customer, use the blacklist feature described above. The App Manager relies on the `osgi/state` folder to "remember" the state of the bundle. If you delete this folder (recommended during patching) the Elasticsearch connector will be reinstalled and started automatically.

Alternatively, use the App Manager.

1. Navigate to Control Panel → Apps → App Manager.

2. Search for *elasticsearch*. Find the Liferay Portal Search Elasticsearch module and click the *edit* (()) button. Choose the *Deactivate* option. This leaves the bundle installed, but stops it in the OSGi runtime.

3. If you're using the Shield and Marvel integration plugins, make sure you uninstall those, too.

Then stop Elasticsearch 2.x. If you're wondering whether your log should be complaining vociferously at this point, the answer is a definitive *yes*. You'll resolve that in the next step.

Install and Configure the Elasticsearch 6 Adapter

Once the default adapter is stopped, install the Elasticsearch 6 adapter (the LPKG you downloaded) by placing it in your Liferay Home folder's `deploy` folder. See here for more information.

Now configure the adapter to find your Elasticsearch cluster by specifying the correct *Cluster Name* and setting *Operation Mode* to *REMOTE*. Make sure the *Transport Address* matches the one Elasticsearch is using. If testing locally with Elasticsearch's default settings, the default value in the adapter works fine (*localhost:9300*).

Reindex

Once the Elasticsearch adapter is installed and talking to the Elasticsearch cluster, navigate to *Control Panel* → *Configuration* → *Server Administration*, and click *Execute* for the *Reindex all search indexes* entry.

You must also re-index the spell check indexes.

Reverting to Elasticsearch 2

Stuff happens. If that stuff involves an unrecoverable failure during the upgrade to Elasticsearch 6, roll back to Elasticsearch 2 and regroup.

Since Elasticsearch 2 and Elasticsearch 6 are two separate installations, this procedure is straightforward:

1. Stop and remove the Elasticsearch 6 adapter.
2. Reinstall the Elasticsearch 2 adapter.
3. Make sure that `elasticsearch.yml` and the Elasticsearch 2 adapter's configuration are configured to use the same port (9200 by default).

Learn more about configuring Elasticsearch in this article.

53.3 Securing Elasticsearch 6.5 with X-Pack

X-Pack is an Elastic extension for securing and monitoring Elasticsearch clusters. If you use Elasticsearch, you should secure it with X-Pack. The security features of X-Pack include authenticating access to the Elasticsearch cluster's data and encrypting Elasticsearch's internal and external communications. These are necessary security features for most production systems. A Liferay Enterprise Search subscription gets you access to both monitoring and security. Contact Liferay's Sales department for more information.

Compatibility: To use X-Pack Security and/or Monitoring with Elasticsearch 6.5 and Liferay DXP, you must use the proper connector to Elasticsearch.

The *Liferay Connector to Elasticsearch 6*, version 1.1.0+ is required to set up Elasticsearch 6.5 with security and monitoring.

Here's an overview of using X-Pack to secure the data indexed in Elasticsearch:

1. Get an Enterprise Search subscription.
2. Configure X-Pack to require authentication and encryption.
3. Download and install the Liferay Enterprise Search Security.
4. Configure the LES Secuirty app with the proper credentials and encryption information.
5. Restart Elasticsearch. These steps require a full cluster restart.

Following these instructions gives you a basic working installation of Elasticsearch communicating freely with Liferay DXP, but read Elastic's documentation to learn about additional configuration options, features, and the architecture of X-Pack.

Setting Up X-Pack Users

In a system using X-Pack Security and X-Pack Monitoring, two of the built-in X-Pack users are important: `kibana` and `elastic`.

Set the passwords for all X-Pack's built-in users. The `setup-passwords` command is the simplest method to set the built-in users' first-use passwords for the first time. To update a password subsequently, use Kibana's UI or the Change Password API.

The `interactive` argument lets you set the passwords for all built-in users. The configuraiton shown in these articles assumes you set all of the passwords to *liferay*. Of course, that's not recommended for production systems.

```
./bin/elasticsearch-setup-passwords interactive
```

See Elastic's documentation on the setup-passwords command for additional options.
Since you're securing Elasticsearch, make sure you keep track of the password set for the `elastic` user.
We recommend enabling transport layer security on each node.

Enabling Transport Layer Security

The following instructions for enabling TLS use `liferay` as the password whenever one is needed. Customize these as appropriate for your installation.

Important: Elasticsearch and Liferay DXP must share the keys and certificates used to configure SSL. Copy them between servers and point to the local copy in the corresponding configuration files.

Generate Node Certificates

Generate a node certificate for each node. You can, of course, use a Certificate Authority toobtain node certificates.

1. Create a certificate authority, using X-Pack's `certutil` command:

   ```
   ./bin/elasticsearch-certutil ca --pem --ca-dn CN=localhost
   ```

   This generates a ZIP file. Unzip the contents somewhere safe.

2. Generate X.509 certificates and private keys using the CA from Step 1. For example:

   ```
    ./bin/elasticsearch-certutil cert --pem --ca-cert /path/to/ca.crt --ca-key /path/to/ca.key --dns localhost --ip 127.0.0.1 --
   name localhost
   ```

   This generates another ZIP file. Extract the contents somewhere in the `Elasticsearch Home/config` folder.

Note: The `certutil` command defaults to using the *PKSC#12* format for certificate generation. Kibana does not work with PKSC#12 certificates, so the `--pem` option (to generate the certificate in PEM format) is important if you're using X-Pack monitoring.

Enable TLS

Enable TLS on each node via its `elasticsearch.yml`.

1. Add the certificate, key and certificate authority paths to each node's `elasticsearch.yml`:

   ```
   xpack.ssl.certificate: /path/to/[Elasticsearch Home]/config/localhost.crt
   xpack.ssl.key: /path/to/[Elasticsearch Home]/config/localhost.key
   xpack.ssl.certificate_authorities: ["/path/to/ca.crt"]
   ```

   The example paths above assume you added the certificate to `Elasticsearch Home/config/`.

2. Enable transport layer TLS with these settings in `elasticsearch.yml`:

   ```
   xpack.security.transport.ssl.enabled: true
   xpack.security.transport.ssl.verification_mode: certificate
   ```

3. Enable TLS on the HTTP layer to encrypt client communication:

   ```
   xpack.security.http.ssl.enabled: true
   ```

After X-Pack is installed and TLS is enabled, configure the LES Security app in Liferay DXP.

Install and Configure the Liferay Enterprise Search Security app

If you have a Liferay Enterprise Search subscription, download the Liferay Enterprise Search Security app. Install the LPKG file by copying it into the `Liferay Home/deploy` folder. That's all there is to it.

To configure the secuirty app, navigate to *Control Panel → Configuration → System Settings*. Find the *Foundation* category and click on the *X-Pack Security* entry. You can enter the property values here, but it's more common to use a configuration file deployed to `Liferay Home/osgi/configs`. Create a file called

```
com.liferay.portal.search.elasticsearch6.xpack.security.internal.configuration.XPackSecurityConfiguration.config
```

The exact contents of the file depend on your X-Pack setup. To configure the adapter according to the Elasticsearch setup documented here, populate the file with these contents:

```
sslKeyPath="/path/to/localhost.key"
sslCertificatePath="/path/to/config/localhost.crt"
certificateFormat="PEM"
requiresAuthentication="true"
username="elastic"
password="liferay"
sslCertificateAuthoritiesPaths="/path/to/ca.crt"
transportSSLVerificationMode="certificate"
transportSSLEnabled="true"
```

The certificate and key files referenced here are the same ones used on the Elasticsearch server. Copy them to the Liferay DXP server and update their paths in the configuration accordingly.

Enable authentication by setting authentication to `required` and providing the credentials for the Elasticsearch user. For SSL, enable transport SSL, set the certificate verification mode and certificate format, and provide the path to the certificate, key, and certificate authority. Of course, the exact values will differ if you configured X-Pack differently.

Here's the complete list of configuration options:

- sslKeyPath
- sslCertificatePath
- sslCertificateAuthoritiesPaths
- certificateFormat
- requiresAuthentication
- username
- password
- transportSSLVerificationMode
- transportSSLEnabled
- sslKeystorePath
- sslKeystorePassword
- sslTruststorePath
- sslTruststorePassword

When you're finished configuring X-Pack Security, restart Elasticsearch. These steps require a full cluster restart.

53.4 Installing X-Pack Monitoring for Elasticsearch 6.5

To monitor Elasticsearch, use X-Pack Monitoring. First install X-Pack onto Elasticsearch and configure security if you're using X-Pack's security features. Then come back here for instructions on installing and configuring Kibana (the monitoring server) with X-Pack so that Elasticsearch, Kibana, and Liferay DXP can communicate effortlessly and securely (if you're using X-Pack Security). A Liferay Enterprise Search subscription is necessary for this integration. Contact Liferay's Sales department for more information.

Compatibility: To use X-Pack Security and/or Monitoring with Elasticsearch 6.5 and Liferay DXP, you must use the proper connector to Elasticsearch.

The *Liferay Connector to Elasticsearch 6*, version 1.1.0+ is required to set up Elasticsearch 6.5 with security and monitoring.

---

1. Tell Elasticsearch to enable data collection.
2. Download and install Kibana.
3. Configure Kibana with the proper security settings.
4. Download and install the Liferay Enterprise Search Monitoring .
5. Configure the connector to communicate with Elasticsearch.

For the X-Pack security procedure, refer to the X-Pack security article.

Enable Data Collection

Monitoring is enabled on Elasticsearch by default, but data collection isn't. Enable data collection by adding this line to `elasticsearch.yml`.

```
xpack.monitoring.collection.enabled: true
```

Now install Kibana.

Install Kibana

Make sure to install the correct version of Kibana. Check the Liferay Enterprise Search compatibility matrix for details.

1. Download Kibana and extract it. The root folder is referred to as *Kibana Home*.
2. Tell Kibana where to send monitoring data by setting Elasticsearch's URL in `kibana.yml`:

   ```
   elasticsearch.url: "http://localhost:9200"
   ```

   If encryption is enabled on Elasticsearch, this is an `https` URL.
3. If not using X-Pack security, start Kibana by opening a command prompt to Kibana Home and entering this command:

   ```
   ./bin/kibana
   ```

If you're using X-Pack's security features on the Elasticsearch server, there's additional configuration required before starting Kibana.

Configure Kibana with Authentication

If X-Pack requires authentication to access the Elasticsearch cluster, follow these steps or refer to Elastic's documentation.

1. Set the password for the built-in `kibana` user in `Kibana Home/config/kibana.yml`:

   ```
   elasticsearch.username: "kibana"
   elasticsearch.password: "liferay"
   ```

   The password is whatever you set it to when initially setting up X-Pack. Once Kibana is installed, you can change the built-in user passwords from the *Management* user interface.
2. If you're not encrypting communication with the Elasticsearch cluster, start Kibana with

   ```
   ./bin/kibana
   ```

   and navigate to `localhost:5601`. Log in with a user who has the `kibana_user` role.

Configuring Kibana with Encryption

Follow these steps to configure Kibana if X-Pack encrypts communication with the Elasticsearch cluster. Consult Elastic's guide for more information.

Add these settings to `kibana.yml`:

```
xpack.security.encryptionKey: "xsomethingxatxleastx32xcharactersx"
xpack.security.sessionTimeout: 600000

elasticsearch.ssl.verificationMode: certificate
elasticsearch.url: "https://localhost:9200"
elasticsearch.ssl.certificateAuthorities: [ "/path/to/ca.crt" ]

server.ssl.enabled: true
server.ssl.certificate: /path/to/[Elasticsearch Home]/config/localhost.crt
server.ssl.key: /path/to/[Elasticsearch Home]/config/localhost.key
```

For more information about monitoring and security best practices in a clustered environment, refer to Elastic's documentation.

After this step you can access Kibana at `https://localhost:5601` and sign in with a Kibana user. The last step is to hook Kibana up with Liferay DXP.

Configuring the Liferay Enterprise Search Monitoring app

If you have a Liferay Enterprise Search subscription, download the Liferay Enterprise Search Monitoring. Install the LPKG file by copying it into the `Liferay Home/deploy` folder. That's all there is to it.

1. Once the connector is installed and Kibana and Elasticsearch are securely configured, create a configuration file named

   ```
   com.liferay.portal.search.elasticsearch6.xpack.monitoring.web.internal.configuration.XPackMonitoringConfiguration.config
   ```

2. Place these settings in the `.config` file:

   ```
   kibanaPassword="liferay"
   kibanaUserName="elastic"
   kibanaURL="http://localhost:5601"
   ```

   Alternatively, configure the monitoring adapter from the Control Panel. Navigate to *Configuration* → *System Settings* and find the X-Pack Monitoring entry in the Foundation category. All the configuration options for the monitoring connector appear there.

   The values differ depending on your Kibana configuration. For example, `kibanaURL="https://localhost:5601"` if using X-Pack Security features.

3. Deploy this configuration file to `Liferay Home/osgi/configs`, and the settings are picked up by your running instance. There's no need to restart the server.

4. There are two more settings to add to Kibana itself. The first forbids Kibana from rewriting requests prefixed with `server.basePath`. The second sets Kibana's base path for the Monitoring portlet to act as a proxy for Kibana's monitoring UI. Add this to `kibana.yml`:

   ```
   server.rewriteBasePath: false
   server.basePath: "/o/portal-search-elasticsearch-xpack-monitoring/xpack-monitoring-proxy"
   ```

   Note that once you set the `server.basePath`, you cannot access the Kibana UI through Kibana's URL (for example, `https://localhost:5601`). All access to the Kibana UI is via the monitoring portlet, which is only accessible to logged in Liferay DXP users. Navigate directly to the portlet using this URL: http://localhost:8080/o/portal-search-elasticsearch-xpack-monitoring/xpack-monitoring-proxy/app/monitoring

5. Because you're using the Monitoring portlet in Liferay DXP as a proxy to Kibana's UI, if you are using X-Pack Security, you must configure the application server's startup JVM parameters to recognize a valid *truststore* and *password*.

   First, navigate to Elasticsearch Home and generate a PKSC#12 certificate from the CA you created when setting up X-Pack security:

```
./bin/elasticsearch-certutil cert --ca-cert /path/to/ca.crt --ca-key /path/to/ca.key --ip 127.0.0.1 --dns localhost --
name localhost --out /path/to/Elasticsearch_Home/config/localhost.p12
```

Next use the `keytool` command to generate a truststore:

```
keytool -importkeystore -deststorepass liferay -destkeystore /path/to/truststore.jks -srckeystore /path/to/Elasticsearch_Home/config/localhost.p12 -
srcstoretype PKCS12 -srcstorepass liferay
```

Add the trustore path and password to your application server's startup JVM parameters. Here are example truststore and path parameters for appending to a Tomcat server's `CATALINA_OPTS`:

```
-Djavax.net.ssl.trustStore=/path/to/truststore.jks -Djavax.net.ssl.trustStorePassword=liferay
```

Restart Liferay DXP and Kibana.

Monitoring in Liferay DXP

Once Kibana and X-Pack are successfully installed and configured and all the servers are up and running, add the X-Pack Monitoring portlet to a page:

1. Open the *Add* menu on a page and choose *Applications*
2. Search for *monitoring* and drag the *X-Pack Monitoring* application from the Search category onto the page.

See the Elastic documentation for information on monitoring Elasticsearch.

CHAPTER 54

SECURING ELASTICSEARCH 6.1 WITH X-PACK

Elasticsearch 6.1 is EOL as of June 13, 2019. If possible, install Elastic stack 6.5 products.

X-Pack is an Elastic extension for securing and monitoring Elasticsearch clusters. If you use Elasticsearch, you should secure it with X-Pack. The security features of X-Pack include authenticating access to the Elasticsearch cluster's data and encrypting Elasticsearch's internal and external communications. These are necessary security features for most production systems. A Liferay Enterprise Search subscription gets you access to both monitoring and security. Contact Liferay's Sales department for more information.

Here's an overview of using X-Pack to secure the data indexed in Elasticsearch:

1. Get an Enterprise Search subscription.
2. Install X-Pack into Elasticsearch and configure it to require authentication and encryption.
3. Download and install the Liferay Enterprise Search Security.
4. Configure the Liferay Enterprise Search Security app with the proper credentials and encryption information.
5. Restart Elasticsearch. These steps require a full cluster restart.

Following these instructions gives you a basic working installation of Elasticsearch communicating freely with Liferay DXP, but read Elastic's documentation to learn about additional configuration options, features, and the architecture of X-Pack.

54.1 Installing X-Pack

1. To install X-Pack and automatically grant it the required permissions (recommended), run

   ```
   bin/elasticsearch-plugin install x-pack --batch
   ```

   on each cluster node. The `--batch` option bypasses installation prompts for granting permissions to X-Pack.

   You'll see log output detailing the permissions granted, finishing with `Installed x-pack`:

```
-> Downloading x-pack from elastic
[=================================================] 100%
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@     WARNING: plugin requires additional permissions     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
* java.io.FilePermission \\.\pipe\* read,write
* java.lang.RuntimePermission accessClassInPackage.com.sun.activation.registries
* java.lang.RuntimePermission getClassLoader
* java.lang.RuntimePermission setContextClassLoader
* java.lang.RuntimePermission setFactory
* java.net.SocketPermission * connect,accept,resolve
* java.security.SecurityPermission createPolicy.JavaPolicy
* java.security.SecurityPermission getPolicy
* java.security.SecurityPermission putProviderProperty.BC
* java.security.SecurityPermission setPolicy
* java.util.PropertyPermission * read,write
See http://docs.oracle.com/javase/8/docs/technotes/guides/security/permissions.html
for descriptions of what these permissions allow and the associated risks.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@        WARNING: plugin forks a native controller        @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
This plugin launches a native controller that is not subject to the Java
security manager nor to system call filters.
Elasticsearch keystore is required by plugin [x-pack], creating...
-> Installed x-pack
```

See more about the permissions X-Pack needs here.

2. Make sure Elasticsearch does not allow the automatic creation of indexes. If you're unsure, check `elasticsearch.yml` for this property:

```
action.auto_create_index: false
```

This property is true by default, so if you don't see it in `elasticsearch.yml`, there's nothing to worry about. See Elastic's documentation for more information on automatic index creation.

3. Restart Elasticsearch.

Once X-Pack is installed, configure its built-in user passwords.

54.2 Setting Up X-Pack Users

In a system using X-Pack Security and X-Pack Monitoring, two of the built-in X-Pack users are important: `kibana` and `elastic`.

Set the passwords for all X-Pack's built-in users. The `setup-passwords` command is the simplest method to set the built-in users' first-use passwords for the first time. To update a password subsequently, use Kibana's UI or the Change Password API.

The `interactive` argument lets you set the passwords for all built-in users. The configuraiton shown in these articles assumes you set all of the passwords to *liferay*. Of course, that's not recommended for production systems.

```
./bin/x-pack/setup-passwords interactive
```

See Elastic's documentation on the setup-passwords command for additional options.

Since you're securing Elasticsearch, make sure you keep track of the password set for the `elastic` user.

We recommend enabling transport layer security on each node.

54.3 Enabling Transport Layer Security

The following instructions for enabling TLS use `liferay` as the password whenever one is needed. Customize these as appropriate for your installation.

Important: Elasticsearch and Liferay DXP must share the keys and certificates used to configure SSL. Copy them between servers and point to the local copy in the corresponding configuration files.

Generate Node Certificates

Generate a node certificate for each node. You can, of course, use a Certificate Authority toobtain node certificates.

1. Create a certificate authority, using X-Pack's `certutil` command:

   ```
   ./bin/x-pack/certutil ca --pem --ca-dn CN=localhost
   ```

   This generates a ZIP file. Unzip the contents somewhere safe.

2. Generate X.509 certificates and private keys using the CA from Step 1. For example:

   ```
    ./bin/x-pack/certutil cert --pem --ca-cert /path/to/ca.crt --ca-key /path/to/ca.key --dns localhost --ip 127.0.0.1 --
   name localhost
   ```

   This generates another ZIP file. Extract the contents somewhere in the `Elasticsearch Home/config` folder.

Note: The `certutil` command defaults to using the *PKSC#12* format for certificate generation. Kibana does not work with PKSC#12 certificates, so the `--pem` option (to generate the certificate in PEM format) is important if you're using X-Pack monitoring.

Enable TLS

Enable TLS on each node via its `elasticsearch.yml`.

1. Add the certificate, key and certificate authority paths to each node's `elasticsearch.yml`:

   ```
   xpack.ssl.certificate: /path/to/[Elasticsearch Home]/config/localhost.crt
   xpack.ssl.key: /path/to/[Elasticsearch Home]/config/localhost.key
   xpack.ssl.certificate_authorities: ["/path/to/ca.crt"]
   ```

   The example paths above assume you added the certificate to `Elasticsearch Home/config/`.

2. Enable transport layer TLS with these settings in `elasticsearch.yml`:

   ```
   xpack.security.transport.ssl.enabled: true
   xpack.security.transport.ssl.verification_mode: certificate
   ```

3. Enable TLS on the HTTP layer to encrypt client communication:

```
xpack.security.http.ssl.enabled: true
```

After X-Pack is installed and TLS is enabled, configure the LES Security app in Liferay DXP.

54.4 Install and Configure the Liferay Enterprise Search Security app

If you have a Liferay Enterprise Search subscription, download the Liferay Enterprise Search Security app. Install the LPKG file by copying it into the `Liferay Home/deploy` folder. That's all there is to it.

To configure security, navigate to *Control Panel* → *Configuration* → *System Settings*. Find the *Foundation* category and click on the *X-Pack Security* entry. You can enter the property values here, but it's more common to use a configuration file deployed to `Liferay Home/osgi/configs`. Create a file called

```
com.liferay.portal.search.elasticsearch6.xpack.security.internal.configuration.XPackSecurityConfiguration.config
```

The exact contents of the file depend on your X-Pack setup. To configure the adapter according to the Elasticsearch setup documented here, populate the file with these contents:

```
sslKeyPath="/path/to/localhost.key"
sslCertificatePath="/path/to/localhost.crt"
certificateFormat="PEM"
requiresAuthentication="true"
username="elastic"
password="GqhoaEUyTM@tp1*wQd~F"
sslCertificateAuthoritiesPaths="/path/to/ca.crt"
transportSSLVerificationMode="certificate"
transportSSLEnabled="true"
```

The certificate and key files referenced here are the same ones used on the Elasticsearch server. Copy them to the Liferay DXP server and update their paths in the configuration accordingly.

Enable authentication by setting authentication to `required` and providing the credentials for the Elasticsearch user. For SSL, enable transport SSL, set the certificate verification mode and certificate format, and provide the path to the certificate, key, and certificate authority. Of course, the exact values will differ if you configured X-Pack differently.

Here's the complete list of configuration options for the LES Security app:

- `sslKeyPath`
- `sslCertificatePath`
- `sslCertificateAuthoritiesPaths`
- `certificateFormat`
- `requiresAuthentication`
- `username`
- `password`
- `transportSSLVerificationMode`
- `transportSSLEnabled`
- `sslKeystorePath`
- `sslKeystorePassword`
- `sslTruststorePath`
- `sslTruststorePassword`

When you're finished configuring X-Pack Security, restart Elasticsearch. These steps require a full cluster restart.

54.5 Installing X-Pack Monitoring for Elasticsearch 6.1

Elasticsearch 6.1 is EOL as of June 13, 2019. If possible, install Elastic stack 6.5 products.

To monitor Elasticsearch, use X-Pack Monitoring. First install X-Pack onto Elasticsearch and configure security if you're using X-Pack's security features. Then come back here for instructions on installing and configuring Kibana (the monitoring server) with X-Pack so that Elasticsearch, Kibana, and Liferay DXP can communicate effortlessly and securely (if you're using X-Pack Security). A Liferay Enterprise Search subscription is necessary for this integration. Contact Liferay's Sales department for more information.

1. Download and install Kibana.
2. Install X-Pack onto Kibana and configure Kibana

   If using X-Pack's security, this includes proper security configuration.
3. Download and install the Liferay Enterprise Search Monitoring [Elastic Stack 6.x].
4. Configure the connector to communicate with Elasticsearch.

The exact steps differ if you're enabling security *and* monitoring or just monitoring. Differences in the process are noted as appropriate.

For the X-Pack installation procedure, refer to the X-Pack security article.

This guide starts with the installation of Kibana.

Install Kibana

Make sure to install the correct version of Kibana. Check the Liferay Enterprise Search compatibility matrix for details.

1. Download Kibana and extract it. The root folder is referred to as *Kibana Home*.
2. Install X-Pack into Kibana:

   ```
   ./bin/kibana-plugin install x-pack
   ```
3. Tell Kibana where to send monitoring data by setting Elasticsearch's URL in `kibana.yml`:

   ```
   elasticsearch.url: "http://localhost:9200"
   ```

   If SSL is enabled on Elasticsearch, this is an `https` URL.
4. If not using X-Pack security, start Kibana by entering

   ```
   ./bin/kibana
   ```

   from Kibana Home.

If you're using X-Pack's security features, there's additional configuration required that you can find below and is related to Securing Elasticsearch 6 with X-Pack.

Configure Kibana with Authentication

If X-Pack requires authentication to access the Elasticsearch cluster, follow these steps or refer to Elastic's documentation.

1. Set the password for the built-in `kibana` user in `Kibana Home/config/kibana.yml`:

   ```
   elasticsearch.username: "kibana"
   elasticsearch.password: "liferay"
   ```

   The password is whatever you set it to when initially setting up X-Pack. Once Kibana is installed, you can change the built-in user passwords from the *Management* user interface.

2. If you're not encrypting communication with the Elasticsearch cluster, start Kibana with

   ```
   ./bin/kibana
   ```

   and navigate to `localhost:5601`. Log in with a user who has the `kibana_user` role.

Configuring Kibana with Encryption

Follow these steps to configure Kibana if X-Pack encrypts communication with the Elasticsearch cluster. Consult Elastic's guide for more information.

Add these settings to `kibana.yml`:

```
xpack.security.encryptionKey: "xsomethingxatxleastx32xcharactersx"
xpack.security.sessionTimeout: 600000

elasticsearch.ssl.verificationMode: certificate
elasticsearch.url: "https://localhost:9200"
elasticsearch.ssl.certificateAuthorities: [ "/path/to/ca.crt" ]

server.ssl.enabled: true
server.ssl.certificate: /path/to/[Elasticsearch Home]/config/localhost.crt
server.ssl.key: /path/to/[Elasticsearch Home]/config/localhost.key
```

For more information about monitoring and security best practices in a clustered environment, refer to Elastic's documentation.

After this step you can access Kibana at `https://localhost:5601` and sign in with a Kibana user. The last step is to hook Kibana up with Liferay DXP.

Configuring the Liferay Enterprise Search Monitoring app

If you have a Liferay Enterprise Search subscription, download the Liferay Enterprise Search Monitoring. Install the LPKG file by copying it into the `Liferay Home/deploy` folder. That's all there is to it.

1. Once the connector is installed and Kibana and Elasticsearch are securely configured, create a configuration file named

   ```
   com.liferay.portal.search.elasticsearch6.xpack.monitoring.web.internal.configuration.XPackMonitoringConfiguration.config
   ```

2. Place these settings in the `.config` file:

```
kibanaPassword="liferay"
kibanaUserName="elastic"
kibanaURL="http://localhost:5601"
```

Alternatively, configure the monitoring adapter from the Control Panel. Navigate to *Configuration* → *System Settings* and find the X-Pack Monitoring entry in the Foundation category. All the configuration options for the monitoring connector appear there.

The values differ depending on your Kibana configuration. For example, `kibanaURL="https://localhost:5601"` if using X-Pack Security features.

3. Deploy this configuration file to `Liferay Home/osgi/configs`, and the settings are picked up by your running instance. There's no need to restart the server.

4. There's one more setting to add to Kibana itself. It sets Kibana's base path to let the Monitoring Portlet act as a proxy for Kibana's monitoring UI. Add this to `kibana.yml`:

```
server.basePath: "/o/portal-search-elasticsearch-xpack-monitoring/xpack-monitoring-proxy"
```

Note that once you set the `server.basePath`, you cannot access the Kibana UI through Kibana's URL (for example, `https://localhost:5601`). All access to the Kibana UI is via the monitoring portlet, which is only accessible to logged in Liferay DXP users. Navigate directly to the portlet using this URL: http://localhost:8080/o/portal-search-elasticsearch-xpack-monitoring/xpack-monitoring-proxy/app/monitoring

5. Because you're using the Monitoring portlet in Liferay DXP as a proxy to Kibana's UI, if you are using X-Pack Security, you must configure the application server's startup JVM parameters to recognize a valid *truststore* and *password*.

   First, navigate to Elasticsearch Home and generate a PKSC#12 certificate from the CA you created when setting up X-Pack security:

```
./bin/x-pack/certutil cert --ca-cert /path/to/ca.crt --ca-key /path/to/ca.key --ip 127.0.0.1 --dns localhost --name localhost --
out /path/to/Elasticsearch_Home/config/localhost.p12
```

   Next use the `keytool` command to generate a truststore:

```
keytool -importkeystore -deststorepass liferay -destkeystore /path/to/truststore.jks -srckeystore /path/to/Elasticsearch_Home/config/localhost.p12 -
srcstoretype PKCS12 -srcstorepass liferay
```

   Add the trustore path and password to your application server's startup JVM parameters. For a Tomcat server, append this to your existing `CATALINA_OPTS`:

```
-Djavax.net.ssl.trustStore=/path/to/truststore.jks -Djavax.net.ssl.trustStorePassword=liferay
```

Restart Liferay DXP and Kibana.

54.6 Monitoring in Liferay DXP

Once Kibana and X-Pack are successfully installed and configured and all the servers are up and running, add the X-Pack Monitoring portlet to a page:

1. Open the *Add* menu on a page and choose *Applications*
2. Search for *monitoring* and drag the *X-Pack Monitoring* application from the Search category onto the page.

See the Elastic documentation for information on monitoring Elasticsearch and monitoring production systems.

Chapter 55

Managing Liferay DXP with Liferay Connected Services

Note: LCS is deprecated and will be shut down on December 31, 2021. Customers who activate LCS are advised to replace it with our latest activation key type which is suitable for virtualized environments.

For further information, please see Changes to Liferay Product Activation.

Liferay Connected Services (LCS) is a set of tools and services for managing and monitoring your Liferay DXP instances. LCS can help you install fix packs, monitor your instances' performance, activate your instances, and help you manage your subscriptions. In other words, LCS is like a butler for the mansion that is Liferay DXP. Even better, the features of LCS work regardless of whether your instance is on a single discreet server or distributed across a cluster. It's like having a single butler that can serve several mansions at once! You can find more information about LCS on its official product page.

Note: You must use LCS for activation of Elastic subscriptions. Otherwise, you don't have to use LCS for activation. You can instead request an XML activation key from Liferay Support.

Before going any further, you should take note of a few key terms used throughout this guide:

Project: Represents a group of users belonging to a company or organization. For example, a project can consist of all the users from a project team or business unit, or it can include the entire company.

Environment: Represents a physical cluster of servers or a virtual or logical aggregation of servers.

Server: Describes a concrete Liferay DXP instance. It can be a standalone server or a cluster node.

As you go through this guide, you'll cover the following topics:

- Getting Started
- LCS Preconfiguration
- Registering Your Liferay DXP Server with LCS
- Using LCS
- Troubleshooting Your LCS Connection

You'll get started with the configuration steps required to use LCS with Liferay DXP.

55.1 LCS Preconfiguration

Note: LCS is deprecated and will be shut down on December 31, 2021. Customers who activate LCS are advised to replace it with our latest activation key type which is suitable for virtualized environments.

For further information, please see Changes to Liferay Product Activation.

Before registering your server with LCS, there are a few things you must configure. The sections in this guide walk you through these steps:

1. Downloading the LCS Client App
2. Preconfiguring LCS to Connect Through a Proxy
3. Ensuring Access to LCS
4. NTP Server Synchronization
5. Configuring the Patching Tool
6. Configuring WebSphere: This is only necessary if you're running Liferay DXP on the WebSphere application server.
7. Installing the LCS Client App

The last section in this guide shows you how to upgrade the LCS client app once your server is registered with LCS. We highly recommend that you upgrade the app whenever Liferay releases a new version of it.

Note: You must use LCS for activation of Elastic subscriptions. Otherwise, you don't have to use LCS for activation. You can instead request an XML activation key from Liferay Support.

Downloading the LCS Client App

The LCS client app is included in each Liferay DXP bundle and autodeploys when the bundle starts. The included version of the app, however, may be outdated. To get the latest version of the LCS client app, you must first download it via Liferay Marketplace.

Note: Even though Liferay Marketplace and this guide use the term *purchase*, the LCS client app is free of charge. The purchase process for a free app in Liferay Marketplace adds the app to your Liferay project, much like downloading a free app in a mobile app store adds the app to your account.

Use these steps to purchase and download the app (if you've already purchased the app, you can skip to step 3 to download it):

1. Navigate to the LCS client app in Liferay Marketplace. Sign in to Marketplace, then click the LCS client app's *Free* button.

2. Select your project, accept the license agreement, and then click the *Purchase* button. Marketplace then displays your receipt.

3. On the receipt, click *See Purchased*. This shows where you can download the LCS client app. To download the app, click the *App* button next to the latest version of the app.

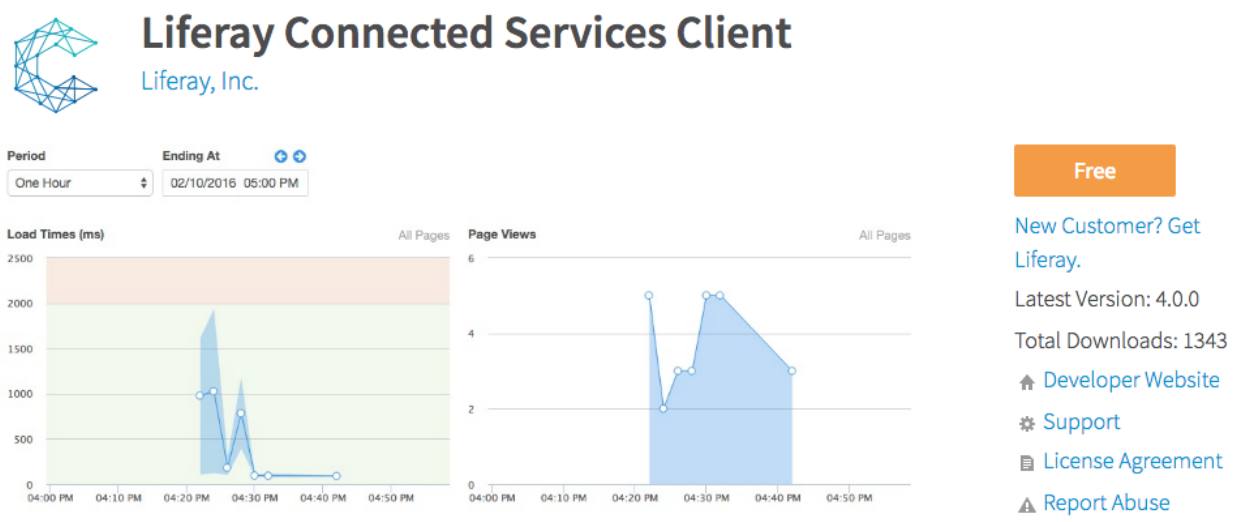


Figure 55.1: Click the app's *Free* button to begin the purchase process.

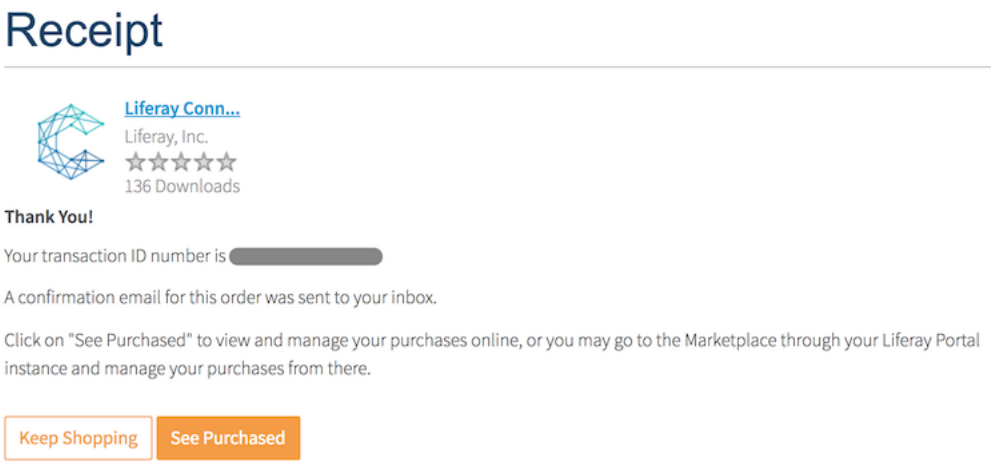


Figure 55.2: Liferay Marketplace displays your receipt for the LCS client app.

```
**Note:** If you must download the LCS client app later, such as when
[upgrading it](#upgrading-the-lcs-client-app),
select *Purchased Apps* from the User menu at the top-right of Liferay
Marketplace. On the Purchased Apps screen, select the project you associated
with the LCS client app and then select the app. This takes you to the same
downloads page shown in the screenshot.
```

```
![ Click the *App* button next to the version of the app you want to download.](./images-dxp/lcs-client-download-page.png)
```

Great! You've successfully downloaded the LCS client app. Before installing it, however, there are a few additional pre-configuration steps you should complete. These appear next; then you'll learn how to install the app.

Note: If your server connects to the Internet through a proxy, you must configure your server or the LCS client app **before** deploying the app. The following section contains instructions on this. If your server doesn't connect through a proxy, skip this section.

Preconfiguring LCS to Connect Through a Proxy

If your server connects to the Internet through a proxy, you must set some properties **before** deploying the LCS client app. There are three ways to do so—chose only one:

1. As JVM app server arguments.
2. As LCS client app properties (LCS client app versions 6+).
3. As LCS client app properties (earlier versions of the LCS client app).

Note: Use only one of these methods to configure your server to connect through a proxy.

JVM App Server Arguments

To set the proxy properties in your server, set them as JVM app server arguments. Set these properties to the appropriate values for your proxy:

```
-Dhttp.proxyHost=
-Dhttp.proxyPort=
-Dhttp.proxyUser=
-Dhttp.proxyPassword=
-Dhttps.proxyHost=
-Dhttps.proxyPort=
```

Note that the `user`, `password`, and `https` properties are only needed if your proxy requires authentication.

LCS Client App Properties: Versions 6+

To set the proxy properties via versions 6+ of the LCS client app, you must create and deploy a config file containing the properties. Follow these steps to do so:

1. Create the config file `com.liferay.lcs.client.configuration.LCSConfiguration.config`. In the steps that follow, you'll set the proxy properties in this file.
2. Set these `proxy*` properties to the appropriate values for your proxy:

   ```
   proxyHostName=""
   proxyHostPort=""
   ```

3. If your proxy requires authentication, pass the credentials via these properties:

   ```
   proxyHostLogin=""
   proxyHostPassword=""
   ```

4. If your proxy requires NTLM authentication, you must also populate these properties:

```
proxyAuthType="ntlm"
proxyDomain=""
proxyWorkstation=""
```

Be sure to set `proxyDomain` and `proxyWorkstation` to the appropriate values for your proxy. Note that you can leave `proxyWorkstation` blank if you don't need it.

5. Deploy the config file to `osgi/configs`.

LCS Client App Properties: Earlier Versions

Versions of the LCS client app prior to version 6 were distributed as a WAR file. To configure these versions of the app to connect through a proxy, you must therefore set their properties in that WAR file. Follow these steps to do so:

1. Extract the LCS client app's WAR file from the app's LPKG file (the app downloads from Liferay Marketplace as an LPKG file). Expand the LPKG file, then locate and expand the client's WAR file: `lcs-portlet-[version].war`.

2. In the LCS client's WAR file, create the file `WEB-INF/classes/portlet-ext.properties` (or open this file if it already exists).

3. Add the following properties at the end of `portlet-ext.properties` and set them to the appropriate values for your proxy:

```
proxy.host.name=
proxy.host.port=
```

4. If your proxy requires authentication, you should also add the following properties and set them to the appropriate values for your proxy:

```
proxy.host.login=
proxy.host.password=
```

5. If your proxy requires NTLM authentication, you must also add the following properties:

```
proxy.auth.type=ntlm
proxy.domain=
proxy.workstation=
```

Be sure to set `proxy.domain` and `proxy.workstation` to the appropriate values for your proxy. Note that you can leave `proxy.workstation` blank if you don't need it.

6. Repackage the LCS client WAR with the modified `portlet-ext.properties` file, then repackage the LPKG file with the LCS client WAR. Make sure the repackaged LPKG file has the same name as the original LPKG file downloaded from Liferay Marketplace.

Ensuring Access to LCS

For the LCS client app to work, it must be able to access the following DNS names. If your server is behind a proxy and/or a firewall, then you must open access to these:

- `lcs.liferay.com`
- `lcs-gateway.liferay.com`

As an added security measure, you can also restrict traffic to HTTPS.
The next section discusses NTP server synchronization.

NTP Server Synchronization

For LCS to work properly, the application server running Liferay DXP should be synchronized with a time server. If it's not, you may get log errors similar to these:

```
ERROR [pool-6-thread-3][HandshakeTask:68] java.lang.RuntimeException: Handshake expired.
Check that the server is synchronized with an NTP server.

WARN [liferay/hot_deploy-1][LCSHotDeployMessageListener:186] LCS portlet is not connected
java.lang.RuntimeException: com.liferay.jsonwebserviceclient.JSONWebServiceInvocationException:
com.fasterxml.jackson.core.JsonParseException: Unrecognized token 'oauth_problem': was expecting
('true', 'false' or 'null')_ at [Source: oauth_problem=timestamp_refused&oauth_acceptable_timestamps=1477311475-1477312075;
line: 1, column: 14] [Sanitized]
```

For information on how to synchronize your application server with a time server, see your application server's documentation.

Next, you'll learn how to configure Liferay DXP's patching tool.

Configuring the Patching Tool

LCS uses Liferay DXP's patching tool to apply updates. In bundles, the patching tool is pre-installed. If you're not running a bundle, you must download and install the patching tool separately.

Once installed, there are a few steps you must complete before LCS can use the patching tool. Note that the commands below apply to Linux, Unix, and Mac. If you're running Windows, drop the `.sh` from each command that has it.

1. Navigate to the `patching-tool` directory on the command line. It's typically located in the Liferay Home folder. Liferay Home is usually the parent folder of the application server's folder.

2. To let the patching tool discover your Liferay DXP installation, run this command:

   ```
   patching-tool.sh auto-discovery
   ```

3. To configure the patching tool, run this command:

   ```
   patching-tool.sh setup
   ```

4. On server startup, the patching tool agent installs the patches downloaded by LCS. For the agent to start with your server, you must set the `javaagent` property in the JVM options to point to your patching tool installation's `patching-tool-agent.jar`. Be sure to specify the correct path to this file:

   ```
   -javaagent:../../patching-tool/lib/patching-tool-agent.jar
   ```

5. If your patching tool is installed in a location other than the Liferay Home folder, you must also specify the `patching-tool` folder's path as an app server JVM argument. Do this with the `patching.tool.home` property:

    ```
    -Dpatching.tool.home=/opt/liferay-dxp-digital-enterprise-7.0/patching-tool/
    ```

There are also a few other things to consider when using the agent. Due to class loading issues, the agent starts in a separate JVM. You can specify options for it with the `patching.tool.agent.jvm.opts` property.

```
-Dpatching.tool.agent.jvm.opts="-Xmx1024m -Xms512m -Dfile.encoding=UTF-8"
```

You may also experience issues on Windows if the user starting the app server doesn't have administrator privileges. Here are some examples of the errors you may see:

```
java.nio.file.FileSystemException: ..\webapps\ROOT\WEB-INF\lib\util-java.jar: Not a file!
java.io.FileNotFoundException: java.io.IOException: Access refused
```

To solve this, set the `java.io.tmpdir` system property as follows in the `patching.tool.agent.jvm.opts` property:

```
-Dpatching.tool.agent.jvm.opts="-Xmx1024m -Xms512m -Dfile.encoding=UTF-8 -Djava.io.tmpdir=%TMP%"
```

The agent also has some flags you can set to control how it behaves:

- `debug`: Provides verbose output in the console.
- `nohalt`: Starts the portal even if the agent encounters an issue.

You can specify these as follows:

```
-Dpatching.tool.agent.properties=debug,nohalt
```

Configuring WebSphere

IBM ® WebSphere ® is a trademark of International Business Machines Corporation, registered in many jurisdictions worldwide.

If you're running the WebSphere application server, then there are some additional configuration steps you must take before deploying the LCS client app:

1. Shut down the application server.

2. Add the following in a `portal-ext.properties` file:

    ```
    module.framework.properties.org.osgi.framework.bootdelegation=\
        __redirected,\
        com.sun.ccpp,\
        com.sun.ccpp.*,\
        com.liferay.aspectj,\
        com.liferay.aspectj.*,\
        com.liferay.portal.servlet.delegate,\
        com.liferay.portal.servlet.delegate*,\
        com.sun.crypto.*,\
        com.sun.image.*,\
        com.sun.jmx.*,\
        com.sun.jna,\
        com.sun.jndi.*,\
        com.sun.mail.*,\
    ```

```
com.sun.management.*,\
com.sun.media.*,\
com.sun.msv.*,\
com.sun.org.*,\
com.sun.syndication,\
com.sun.tools.*,\
com.sun.xml.*,\
com.yourkit.*,\
com.ibm.*,\
sun.*
```

3. In your Liferay DXP installation, delete the `osgi/state` folder.

4. Start the application server.

5. Navigate to the WebSphere console in a browser.

6. Select your server and navigate to *Java and Process Management → Process Definition → Additional Properties*.

7. Select *Java Virtual Machine → Custom Properties*.

8. Click *New*, and enter the following:

    - Name: `com.ibm.crypto.provider.DoRSATypeChecking`
    - Value: `false`

9. Click *Save*, then *OK* to apply changes to the master configuration.

Note that for LCS client app versions prior to 5.0.0, you must also change the value of the `digital.signature.algorithm.provider` property in the app's `portlet.properties` file to `IBMJCE`:

```
digital.signature.algorithm.provider=IBMJCE
```

Installing the LCS Client App

Once you've addressed the above pre-configuration steps, you're ready to install the LCS client app. Follow these steps to install the app:

1. In your Liferay Home folder (usually the parent folder of the application server's folder), delete this file:

    ```
    osgi/marketplace/Liferay Connected Services Client.lpkg
    ```

2. Place the new `Liferay Connected Services Client.lpkg` in `osgi/marketplace`.

Great! Now you're all set to register your server with LCS.

The next section shows you how to upgrade the LCS client app. We highly recommend that you do this whenever Liferay releases a new version of the app.

Upgrading the LCS Client App

Your server should always be running the latest version of the LCS client app. There are two ways to upgrade the app, depending on the exact LCS pre-configuration steps you followed:

1. Via Liferay Marketplace *inside* Liferay DXP. Use this method if you don't need to configure the LCS client app (e.g., to connect through a proxy) before it deploys.

```
**Note:** If you choose this method and have a clustered environment, you
must perform the upgrade separately on each node in your cluster. Therefore,
you may prefer to upgrade manually as detailed in the next step to ensure
that all your nodes are running the exact same version of the LCS client
app.
```

```
To perform the upgrade, first navigate to *Control Panel* &rarr; *Apps*
&rarr; *Purchased*. Apps needing an update are listed first. Click *Update*
next to the LCS client app. Note that you may need to restart your server
for the upgrade to complete.
```

2. Manually, after downloading the LCS client app's LPKG file to your machine. Use this method if you must pre-configure the LCS client app to connect through a proxy.

```
**Note:** If you used JVM app server arguments to configure your server to
connect through a proxy, then you don't need to pre-configure the LCS client
app to connect through the same proxy.
```

```
To update the LCS client app manually, follow the previous sections in this
guide for downloading and pre-configuring the app. Then deploy it to
`[Liferay Home]/deploy` as you would any other app.
```

Contact Liferay Support if you need additional assistance with the upgrade process.

55.2 Registering Your Liferay DXP Server with LCS

Note: LCS is deprecated and will be shut down on December 31, 2021. Customers who activate LCS are advised to replace it with our latest activation key type which is suitable for virtualized environments.

For further information, please see Changes to Liferay Product Activation.

Follow these steps to register your Liferay DXP server with LCS:

1. Ensure that you've completed the LCS preconfiguration steps.

2. Log in to lcs.liferay.com. This takes you to your company's LCS project. If your company has multiple projects, from the menu to the right of the Dashboard tab select the project that's getting a new server.

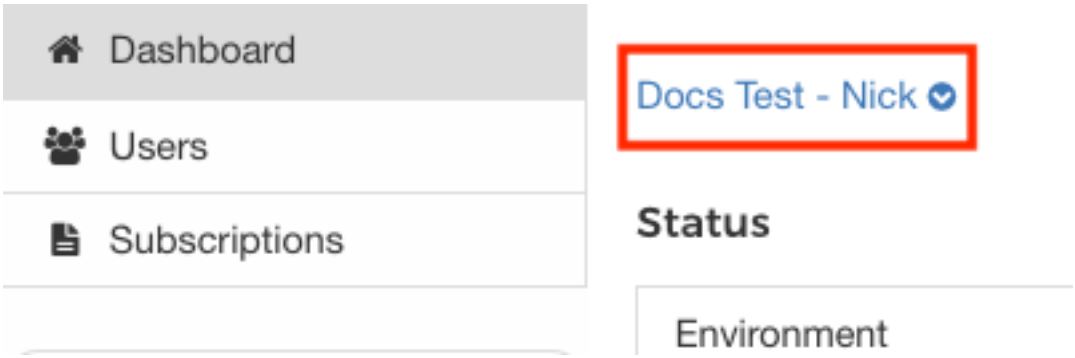


Figure 55.3: Select your LCS project from the menu highlighted by the red box in this screenshot.

3. Select or create the environment in which to register this server. If you're using LCS for activation, upon connection to LCS your server consumes an activation key from the subscription type assigned to the environment. Note that a subscription type can only be assigned to an environment when creating the environment. If you have sufficient permissions in your company's project, you can create a new environment by selecting *Add Environment*.

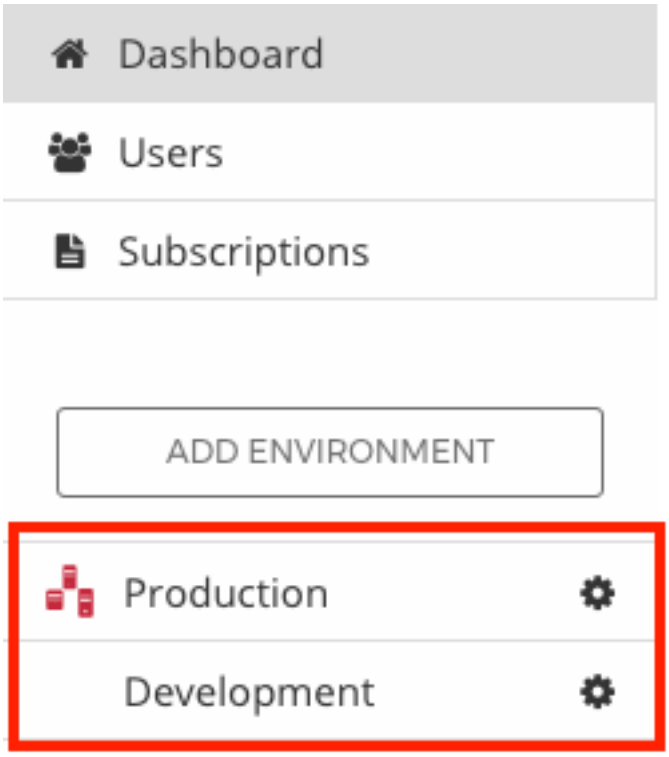


Figure 55.4: You must register your server in an LCS environment. The red box in this screenshot highlights environments.

4. Select the environment's *Registration* tab. This is where you manage and download the environment's token file, that registers servers in the environment.

   In the Registration tab's *Services* section, change the LCS service selections, if needed. Note that if you change the LCS service selections and there are servers already registered in the environment, you

must regenerate the token file and use it to reconnect those servers to LCS. You'll regenerate and/or download the token in the next step.

The selected services are enabled for all servers that connect to this environment. If Portal Property Analysis is selected, you can prevent LCS from analyzing specific properties. Enter them into the box that appears when you select *Show Blacklisted Properties*. Note that LCS doesn't access security sensitive properties.

5. What you do now depends on what you did in the previous step:

    - **Changes to LCS service selections:** Regenerate and download the token. Regenerating a token causes all servers using the old token to disconnect from LCS. You must reconnect them using the new token.
    - **No changes to LCS service selections:** Download the token.

6. Place the token file in your server's `[Liferay Home]/data` folder. Note that Liferay Home is usually the parent folder of the application server's folder. If your server is running, it should connect to LCS in about one minute. If your server isn't running, it connects to LCS on startup.

7. Celebrate! Your Liferay DXP server is registered in LCS. If for some reason it isn't, see the LCS troubleshooting article.

Note: You may be wondering what happens if LCS goes offline. Don't worry, this doesn't cause a rift in the space-time continuum. LCS is deployed on a global cloud infrastructure set up for automatic failure recovery. The potential for non-availability is very low. In the event of an outage, however, registered servers maintain a local copy of their uptime information to transmit to LCS when it comes back online. If you use LCS for activation, active subscriptions have a 30-day grace period to re-establish connectivity and remain valid. This is ample time for LCS to come back online.

Determining Your Server's LCS Connection Status

In Liferay DXP, you can view your LCS connection status in the LCS client app. Access the client by clicking *Control Panel → Configuration → Liferay Connected Services*.

Here's a full description of what a connected LCS client app displays:

Connection Uptime: The duration of the client's connection with LCS.

Last Message Received: The time the LCS client received the latest connection message from LCS. These messages occur only upon connection/reconnection and are unrelated to server metrics. It's therefore common for a long period of time to pass before the client receives another such message for a reconnection event.

Services: The LCS services enabled for this server. Note that all servers in an environment use the same set of LCS services. LCS services can't be controlled on a server-by-server basis.

Project Home: A link to this server's LCS project.

Environment: A link to this server's LCS environment.

Server Dashboard: A link to the server on LCS.

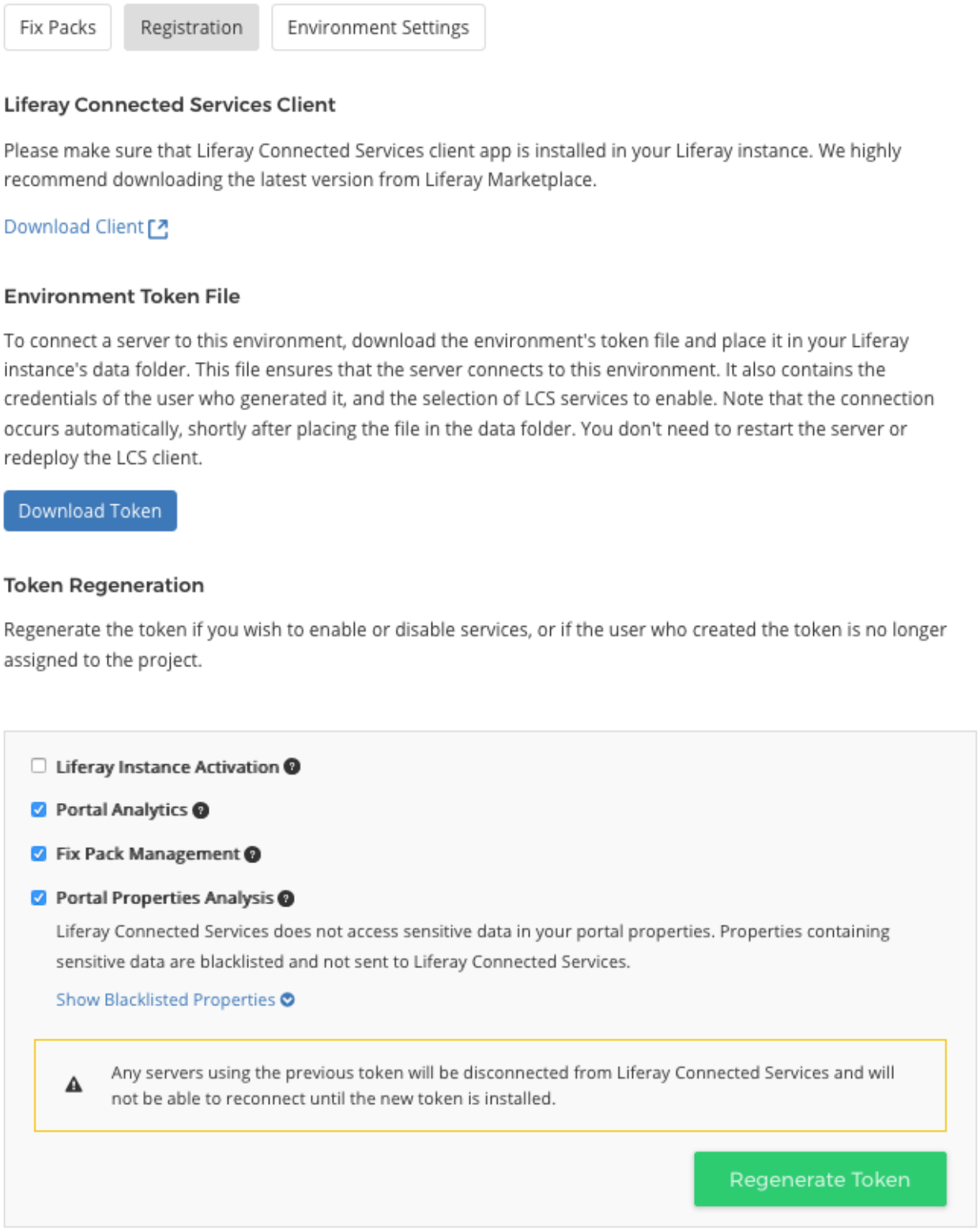


Figure 55.5: An environment's Registration tab lets you manage the token file used to register your server in the environment.

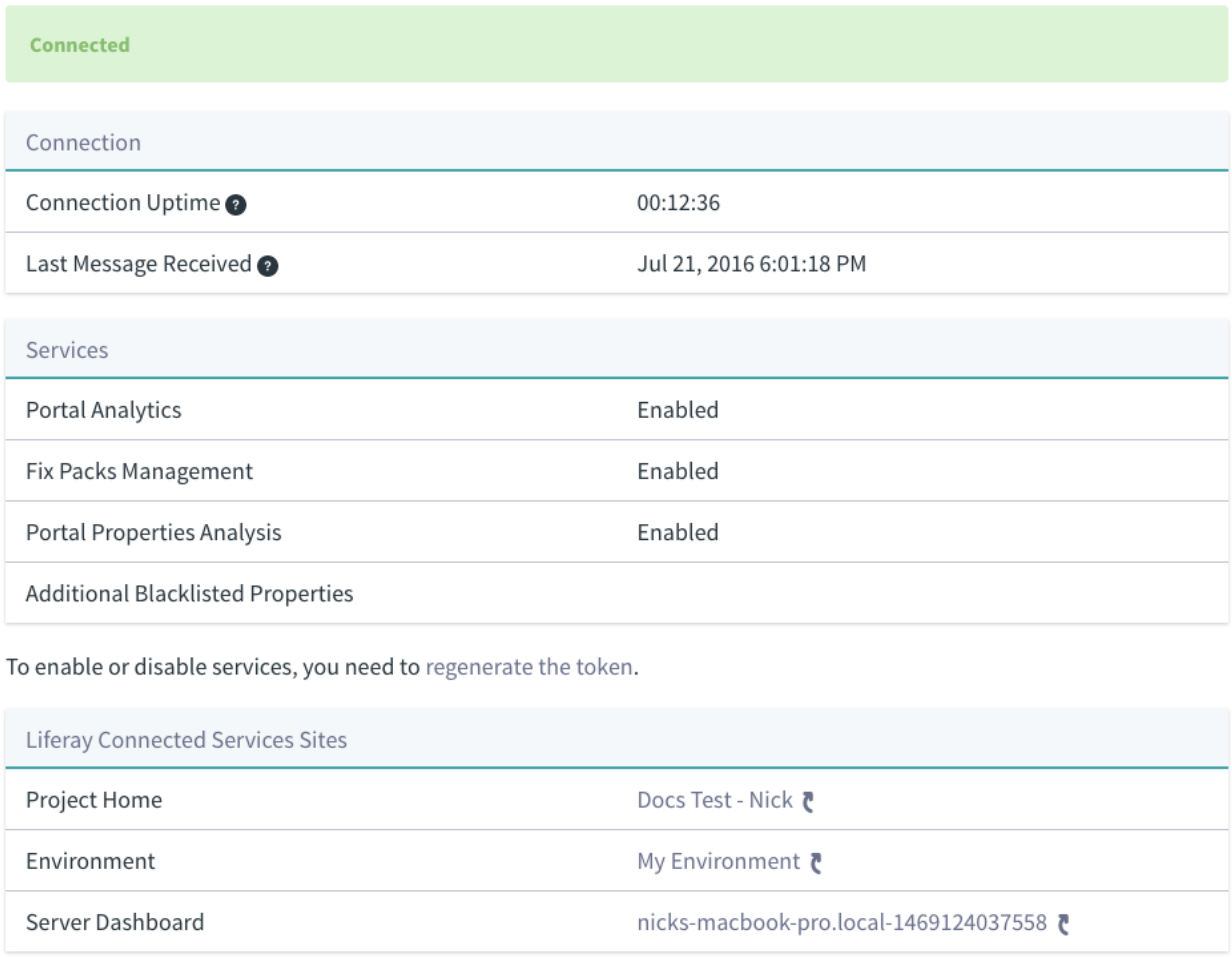


Figure 55.6: The server is connected to LCS.

55.3 Using LCS

Note: LCS is deprecated and will be shut down on December 31, 2021. Customers who activate LCS are advised to replace it with our latest activation key type which is suitable for virtualized environments.

For further information, please see Changes to Liferay Product Activation.

Once your Liferay DXP server is connected to LCS, you can get down to the business that LCS is designed for—managing and monitoring your servers. If you're not already there, log in with your account on lcs.liferay.com. This is where you'll apply updates, view server metrics, manage environments, and more.

This article's sections each detail one or more of LCS's features:

- **What LCS Stores About Your Liferay DXP Servers:** For LCS to work, the LCS servers must store certain information about your servers. Sensitive data, however, isn't stored on the LCS servers. This section describes the data that LCS does and doesn't store.

- **Managing LCS Users in Your Project:** Learn how to manage your LCS project's users by assigning them roles.
- **Using the Dashboard:** Learn how to manage your LCS projects and access your environments and servers in LCS.
- **Managing LCS Environments:** Learn how to create and manage your LCS project's environments. This includes instructions on installing fix packs for an environment's servers.
- **Managing LCS Servers:** Learn how to manage your servers in LCS. This includes viewing server metrics and editing server settings.
- **Managing Your LCS Account:** Learn how to manage your LCS account. This includes setting general account preferences, managing LCS web notifications, and configuring LCS to send you notification emails when specific events occur in your LCS projects.
- **Managing Liferay DXP Subscriptions:** Learn how to view and manage your Liferay DXP subscriptions for the servers in your LCS project.
- **Understanding Environment Tokens:** Learn about the environment tokens that you use to connect your servers to LCS.

What LCS Stores About Your Liferay DXP Servers

To protect your users' privacy, LCS only stores system-specific data. LCS doesn't gather or store data on your users.

By default, LCS stores the following information about your server:

- Portal build number and edition
- Patching Tool Version
- LCS Client Build Number
- Application Server Name
- Database Name
- File Encoding
- OS Name and Version
- Timezone
- IP Address
- Java Version and Java Options
- Number of Processor Cores
- File System Usage
- Memory Usage

The other data LCS stores depends on the services you enable in your environment token. For more information on this, see Registering Your Liferay DXP Server with LCS. When you enable the following services, LCS gathers and stores the data listed for each:

- **Portal analytics:**
  - Portal and portlet metrics
  - JVM metrics
  - Cache and server metrics

- **Fix pack management:**
  - Patches installed on the server
- **Portal properties analysis:**
  - `portal.properties` (except sensitive data)

Sensitive data is any key-value pair that contains user names or passwords. For example, LCS doesn't store the following properties because they contain sensitive data:

```
omniadmin.users
ldap.security.credentials.0, ldap.security.credentials.1, ldap.security.credentials.2 ...
facebook.connect.app.secret
auth.token.shared.secret
auth.mac.shared.key
captcha.engine.recaptcha.key.private
amazon.secret.access.key
tunneling.servlet.shared.secret
microsoft.translator.client.secret
dl.store.s3.secret.key
auto.deploy.glassfish.jee.dm.passwd
```

LCS also doesn't store properties that end in `.password`, besides the following non-sensitive properties:

```
portal.jaas.plain.password
portal.jaas.strict.password
login.create.account.allow.custom.password
```

LCS also lets you prevent it from analyzing specific properties of your choosing. For more information on this, see Registering Your Liferay DXP Server with LCS.

Managing LCS Users in Your Project

The Users section of LCS is where you manage the LCS users that are part of your project. It's here that you can grant or revoke LCS Roles. To manage users, first click the *Users* tab just below the Dashboard tab on the upper-left of your screen.

Note: You can't add users to your project via the LCS UI or the LCS client app. To add users to your project, you must contact Liferay support.

The *Users* tab displays a list of the users in your project. This list includes each user's name, email, image, LCS Roles, and a *Manage Roles* button. Each LCS user must have an assigned Role. The following Roles are available:

LCS Administrator: All LCS functionality is available to administrators. This is the only Role that can manage other users' Roles.

LCS Environment Manager: All LCS functionality is available in the scope of an environment, with the exception of managing other users.

LCS Environment Viewer: Has read-only access in the scope of an environment.

You should note that each of these LCS Roles assume users already have the LCS User Role in their Liferay.com accounts. The LCS User Role is granted automatically the first time a user logs into LCS. The actions that can be performed by each of the LCS Roles are detailed in the below permissions matrix.

LCS Permissions Matrix

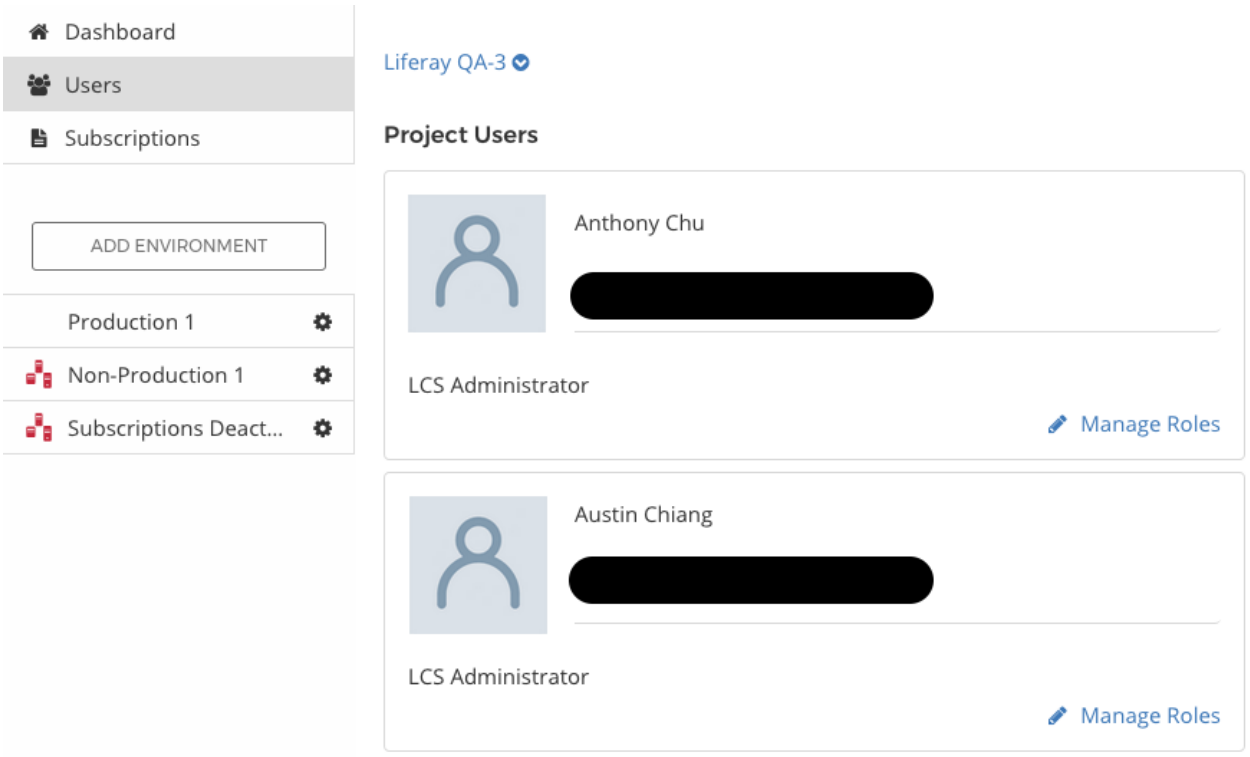


Figure 55.7: The Users tab lets you manage the LCS users in your project.

Action | LCS Administrator | LCS Environment Manager | LCS Environment Viewer | Access LCS | true | true | true | Access Any Environment | true | false | false | Access a Particular Environment | true | true | true | Manage Users | true | false | false | Create and Delete Environments | true | false | false | Edit Any Environment | true | false | false | Edit a Particular Environment | true | true | false | Server Registration in Any Environment | true | false | false | Server Registration in a Particular Environment | true | true | false | Install Fix Packs in Any Environment | true | false | false | Install Fix Packs in a Particular Environment | true | true | false |

Now that you know what Roles are available in an LCS project and what they do, you're ready to learn how to manage them.

Managing LCS Roles

Follow these steps to manage a user's LCS Roles:

1. Click the user's *Manage Roles* button.
2. To revoke a Role, click *Revoke Role* for that Role.
3. To assign a Role, choose the Role (and environment, if applicable) and click *Assign*.

Note: A user can't have an environment Role (e.g., LCS Environment Manager, LCS Environment Viewer) and the LCS Administrator Role at the same time.

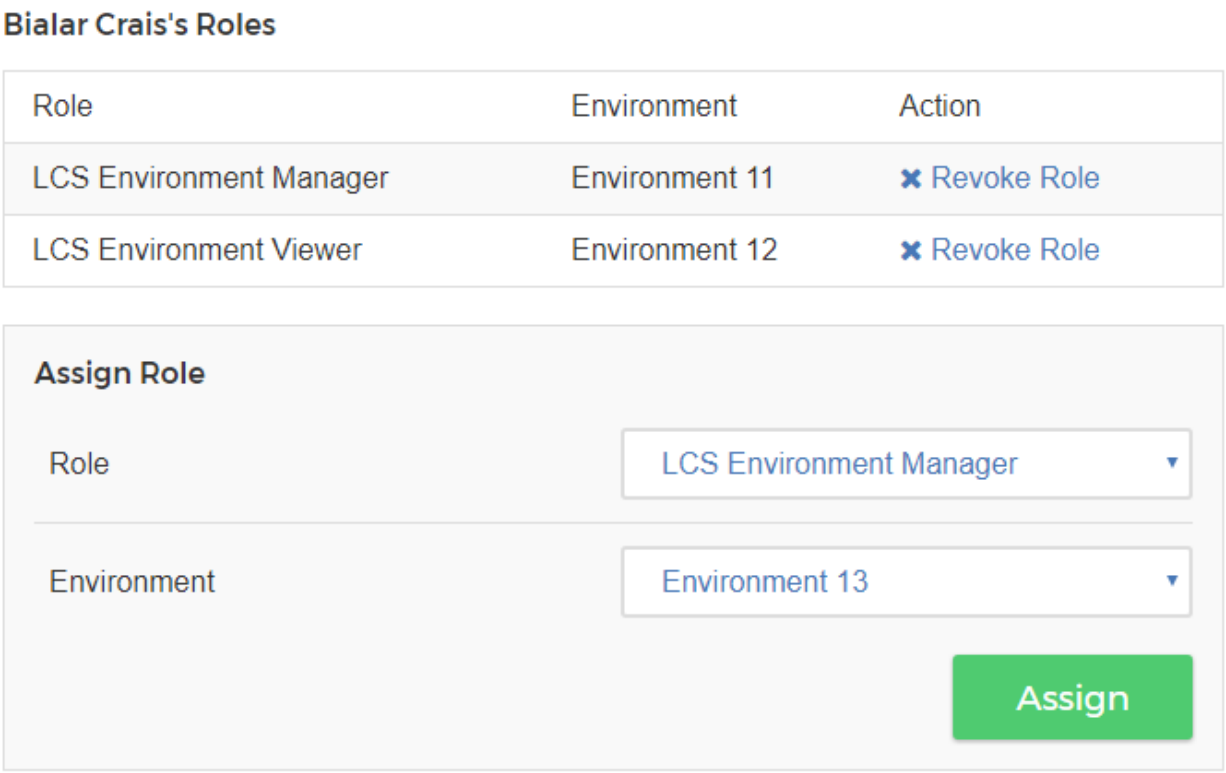


Figure 55.8: You can assign or revoke a user's LCS Roles.

Using the Dashboard

The LCS Dashboard shows a project's environments and servers. If you're not already at the Dashboard, click it near the upper left-hand corner of your LCS site. Clicking *Dashboard* takes you to the project view. From there, you can get to the environment view and the server view. Each of these views gives you a different look into certain aspects of your LCS project. You'll start with the project view.

Using the Project View

You can get to the project view at any time by clicking the *Dashboard* tab near the upper left-hand corner of your LCS site. The project appears to the right of this tab, with a drop-down arrow for switching between projects if you have more than one. You can also switch between projects from the user menu at the top right of the Dockbar. The project view contains a Status table that lists status messages for each server in your project. For example, a status message appears for a server when the server is offline. Status messages also appear for servers when fix packs are available, monitoring is unavailable, the patching tool is unavailable, or other events occur that relate to LCS.

LCS lists the environments in your project on the left side of the screen. You can also create new environments here by clicking the *Add Environment* tab (more on this shortly). To view an environment's settings, click the environment's gear icon. Clicking an environment shows more information about it. This takes you to the environment view. Also note that each environment's icon indicates the environment's type and status:

- **Red icon:** There's a problem with one or more of the environment's servers.

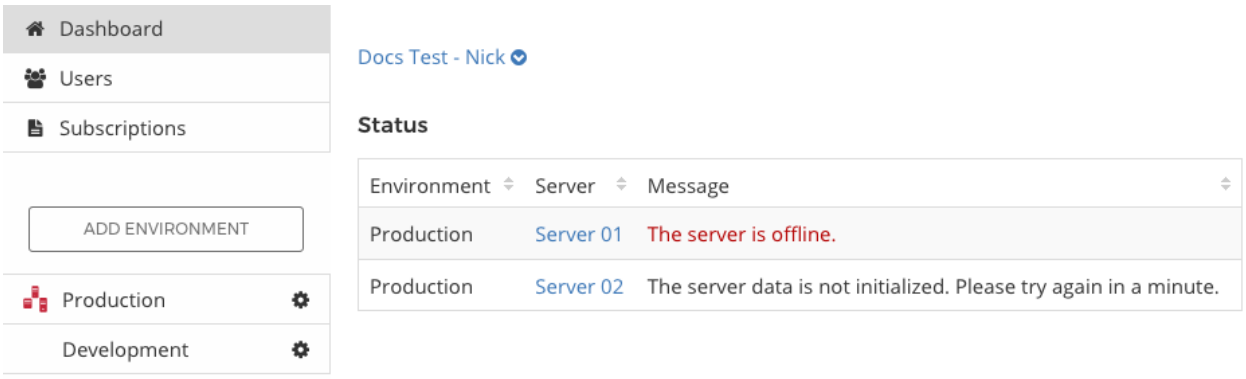


Figure 55.9: The LCS project view shows an overview of your LCS project.

- **Green icon:** The environment's servers are operating properly.
- **Icon with a circle:** The environment's servers are clustered.

Managing LCS Environments

Environments are the key components of your LCS project. When you register a server in LCS, you do so in an environment. An environment is therefore the gateway to managing and monitoring your servers in LCS.

Working with Environments

Clicking an environment on the left-hand side of the Dashboard takes you to the environment view, which lets you manage an environment in your LCS project.

The UI is segmented into three tabs:

1. **Fix Packs:** View and apply fix packs for the environment's servers. This tab only appears if a server is registered in the environment. A table displays the following information for each fix pack:

    - **Name:** The fix pack's name.
    - **Status:** The fix pack's status.
    - **Server:** The server the fix pack can be applied to.
    - **Size:** The fix pack's size. This only appears if the server is running.
    - **Download:** A button to download the fix pack to the server. This only appears if the server is running.

    Once a fix pack downloads, LCS prompts you to restart your server, which installs any downloaded fix packs. Note that you must start your server with the privileges required to write to the disk location where patches are stored and processed (the `patching-tool` folder). To use LCS to install fix packs across a cluster, follow the same procedure. LCS downloads and installs fix packs simultaneously across all nodes—you don't have to handle each separately.

2. **Registration:** Generate and download *environment tokens* that connect your servers to LCS.

3. **Environment Settings:** Change the environment's name, location, and description. You can also see if the environment allows clustered servers and view the environment's subscription type. Click the *Save* button to save any changes you make in the Environment Settings tab. You can also delete the environment by clicking *Delete Environment*.

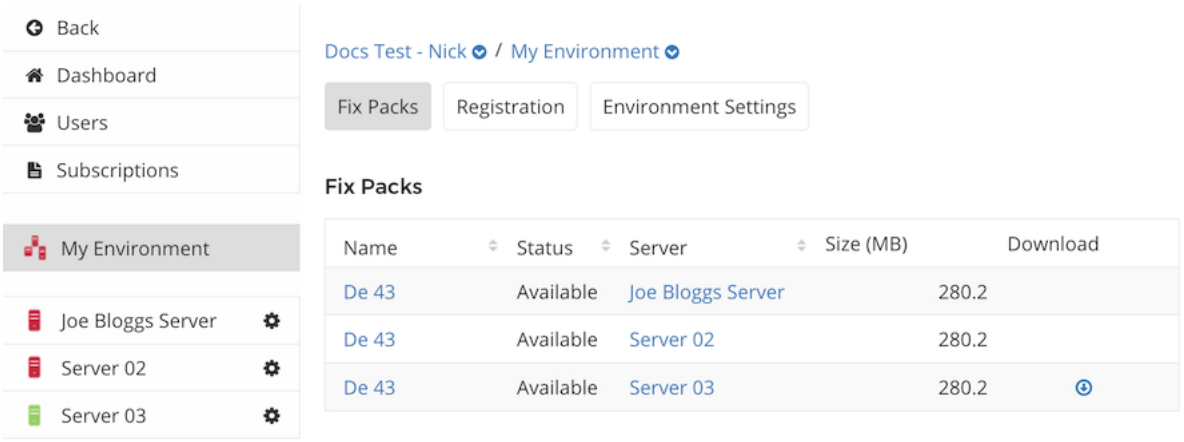


Figure 55.10: The LCS environment view shows an overview of an LCS environment.

Managing LCS Servers

Clicking a server in the Dashboard or environment view takes you to the server view. Server view provides detailed information about a server, including statistics and performance metrics. To protect your users' privacy, LCS doesn't gather, store, or analyze user data.

Server view is segmented into six tabs:

Page Analytics: Displays metrics on page views and load times.

Snapshot Metrics: Displays application, JVM, and server metrics.

Fix Packs: Displays the server's available and installed fix packs.

Portal Properties: Displays your portal's properties and their settings.

Details: Displays general information about your Liferay DXP installation, Java version, and hardware.

Server Settings: View or change your server's name, location, and description. You can also unregister the server from LCS.

Note: LCS only supports Snapshot Metrics for servers running on Tomcat or WebLogic. On other application servers you may see a console message indicating that LCS doesn't support server metrics for your application server. You may also see a benign `NullPointerException` for the LCS `TaskSchedulerServiceImpl` and `ScheduleTasksCommand`.

Page Analytics

Page Analytics appears by default when you enter server view. Page Analytics shows page views and load times for the selected site and time period. By default, all sites are selected. You can select a specific site from the *Site* selector menu. You can also select a different time period in the *Period* and *Ending At* fields. The arrows next to the *Ending At* field move the selected time period up or down, respectively, by one period. For example, if you select *One Hour* in the *Period* field, pressing the right arrow next to *Ending At* moves the selected time period up by one hour. Note that at the beginning of the current time period, it can take up to 15 minutes for data to become available. Also note that data is available for three months from the time LCS collected it.

By default, load times and page views for all pages are plotted against time in separate graphs. Below these graphs, a table displays summary statistics of data over the same time period, for each page. If you click a page in the table, the graphs plot the data for just that page. If you can't find the page you're looking

for, you can search for it in the *Search* box at the top of the table. To plot data for all pages again, click the *All Pages* row at the bottom of the table.

Load times are also color coded to indicate speed. The *Load Times* graph's background is red for values above 3,000 ms, orange for values from 2,000 to 3,000 ms, and green for values less than 2,000 ms. Likewise, the table displays all load times greater than 3,000 ms in red text.

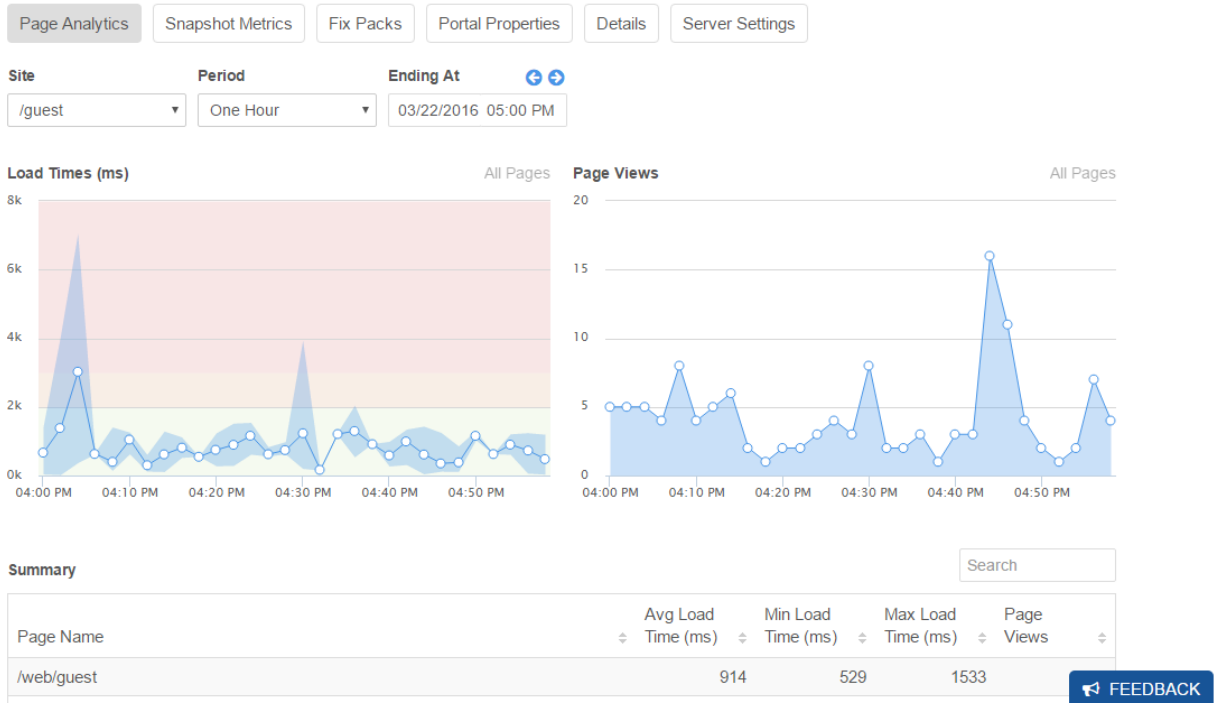


Figure 55.11: The Page Analytics interface in the LCS Server view.

Snapshot Metrics

To view other metrics and statistics of your server's performance, click the *Snapshot Metrics* tab near the top of the page. These metrics are broken down into three main categories: *Application*, *JVM*, and *Server*. Application is selected by default when you click the Snapshot Metrics button.

The Application category also has three other categories: *Pages*, *Portlets*, and *Cache*. Pages lists the frequency that specific pages load, along with their average load times. Portlets lists the same statistics, but for specific portlets in your server. The Cache category lists Liferay Single VM metrics and Hibernate metrics. The following screenshot shows the statistics in the Portlets category.

The JVM category, as its name indicates, shows statistics about the JVM running on your server. This includes data on the garbage collector and memory. The number of runs, total time, and average time are listed for each garbage collector item. The memory metrics are presented in a bar chart that shows the usage of the PS Eden Space, Code Cache, Compressed Class Space, PS Old Gen, PS Survivor Space, and Metaspace.

Server is the third category in Snapshot Metrics. The Server category shows additional information about how your server is running. For example, horizontal bar graphs show the number of current threads running on your server, as well as the JDBC connection pools.

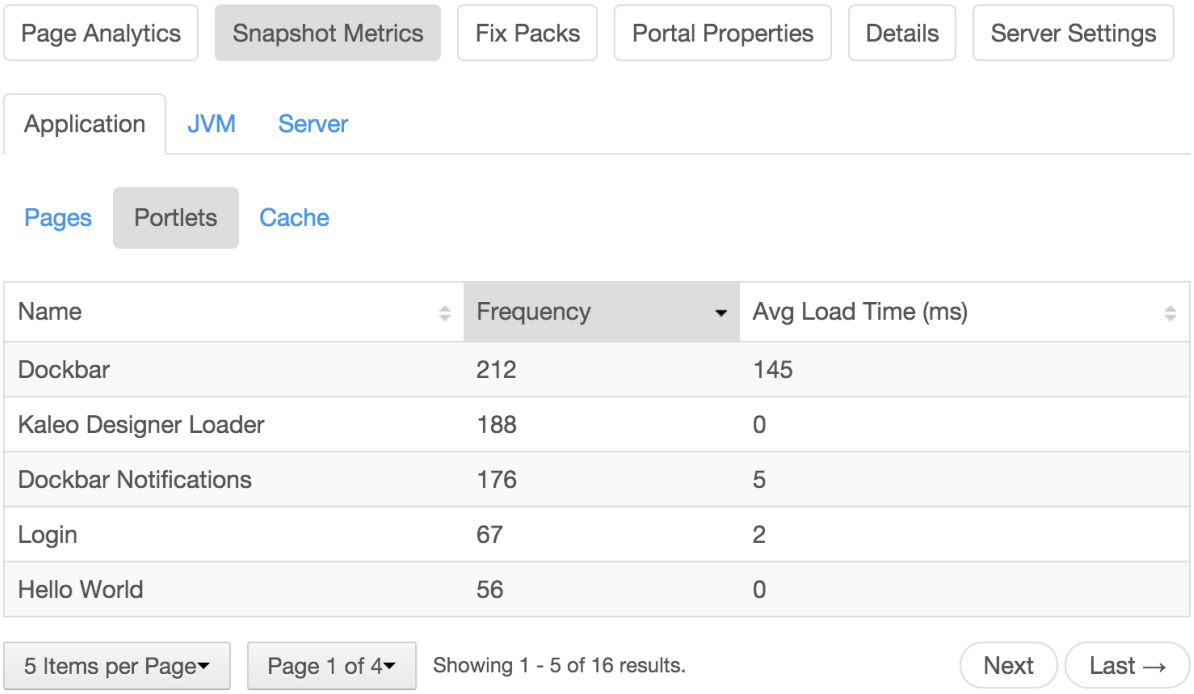


Figure 55.12: The LCS application metrics show portlet performance statistics, like frequency of use and average load time.

Note that in Snapshot Metrics, the application and garbage collector metrics are based on data collected by LCS from server registration to the present. Memory and server metrics, however, show only the current state.

Fix Packs

To view your server's fix packs, click the Fix Packs tab near the top of the page. The available and installed fix packs appear in separate tables. The available fix packs table functions exactly like the Fix Packs table in environment view for downloading and installing fix packs.

Portal Properties

The *Portal Properties* tab lets you view your portal's property values in a searchable table. This gives you a convenient display for your portal property settings. The properties in this table are organized into the following categories:

Default Values: The default values for your portal's properties.

Custom Values: Any custom values you've set for your portal's properties. This includes any property values you change via a `portal-ext.properties` file.

Dynamic Properties: Any property values set at runtime. For example, the Liferay Home folder's location depends on your configuration. To specify this folder when setting any properties that require it, use `${liferay.home}` instead of an absolute directory path.

You can display any combination of these categories by selecting the corresponding checkboxes from the gear icon next to the search box at the top-right of the table. For example, by checking the *Show Default Values*

Application | JVM | Server

Garbage Collector Metrics

Name	Runs	Total Time (ms)	Average Time (ms)
PS MarkSweep	6	2242	373
PS Scavenge	175	2378	13

Memory Metrics

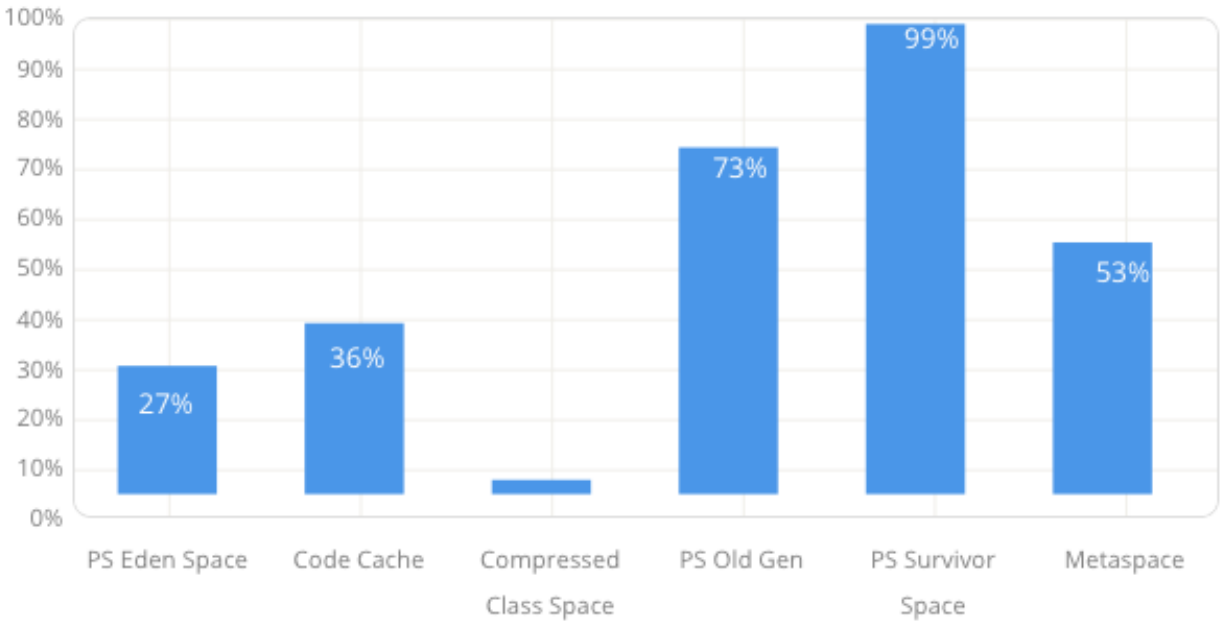


Figure 55.13: The LCS JVM metrics show performance data for memory and the garbage collector.

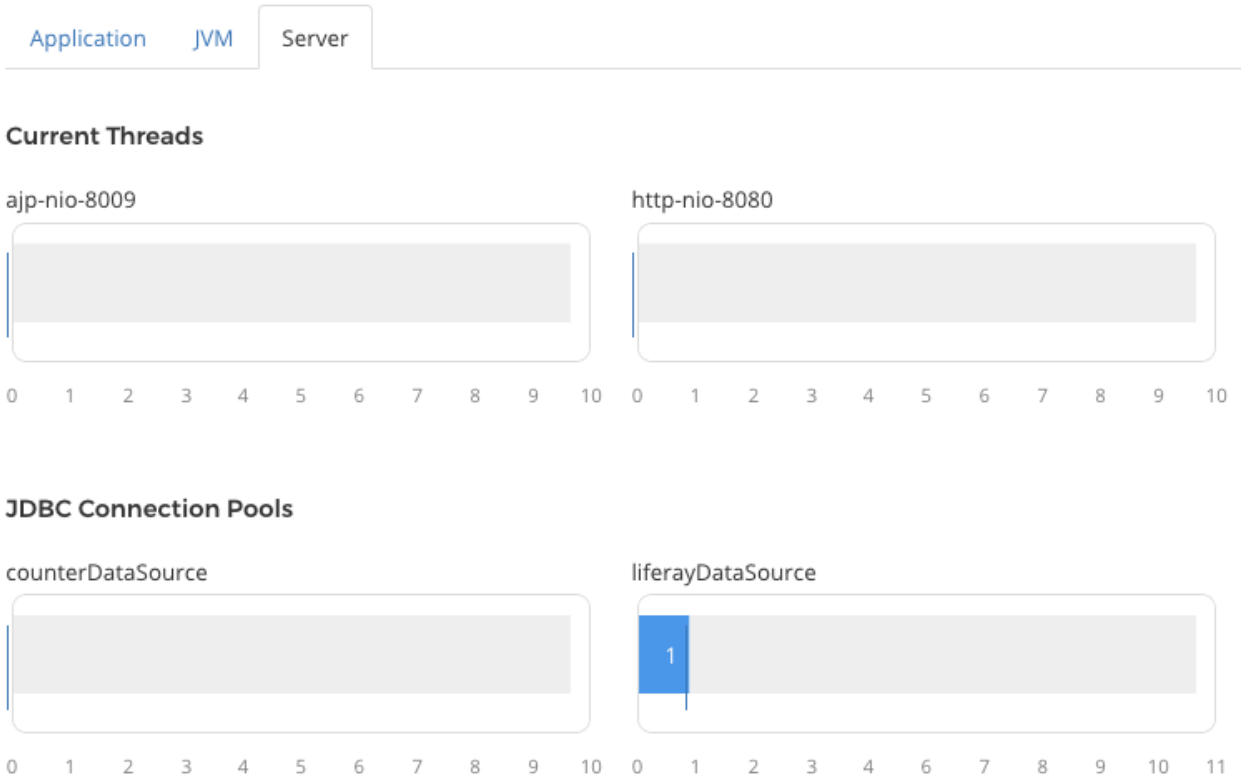


Figure 55.14: The LCS server metrics show current threads and JDBC connection pools.

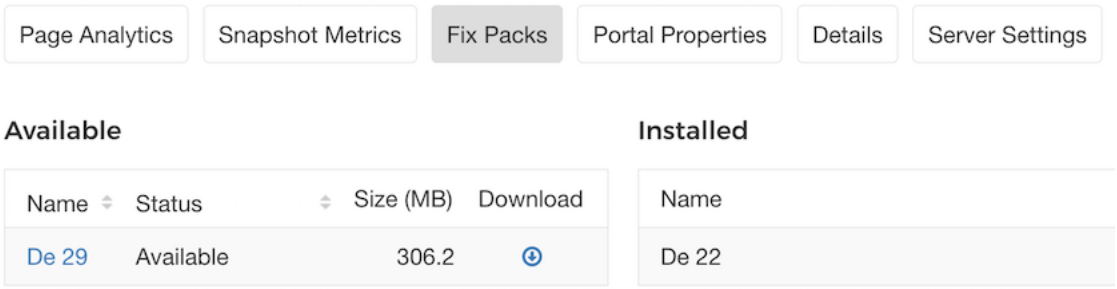


Figure 55.15: The Fix Packs tab displays your server's available and installed fix packs.

and *Show Custom Values* checkboxes, the table shows your portal's default and custom property values. To show only the custom values, select only *Show Custom Values*.

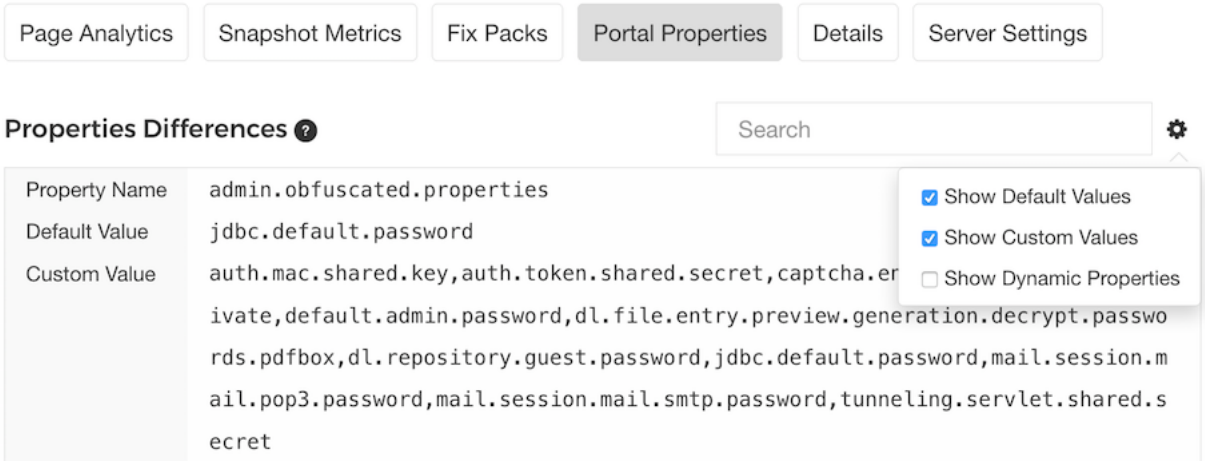


Figure 55.16: : Click the gear icon to select the type of portal properties to show in the table.

Details

The *Details* tab shows general information about your server. There are three tabs under Details: *Software*, *Java*, and *Hardware*. Each shows information, respectively, about your Liferay DXP installation, Java installation, and hardware. This information is useful to the Liferay Support team in the event you need their assistance.

Server Settings

Finally, the *Server Settings* tab lets you view and edit your server's name, location, and description. You can also unregister your server from LCS.

Managing Your LCS Account

To manage your LCS account, select *My Account* from the user menu in the Dockbar. This takes you to a UI with four tabs:

Projects: Displays your LCS projects in a searchable table that lists the administrator's email address for each project.

Email Notifications: Configure LCS to send you emails when specific events occur in your LCS projects by adding rules that define what events trigger a notification. There are no rules by default. Click the *Add Rule* button to define one.

First specify the project, environment, and server for the notification. Note that you have the option of selecting all environments and servers in a project. Then check the checkbox for each event that you want to trigger an email notification. For example, if you create a notification rule with *The server unexpectedly shuts down* selected for all servers and environments in your project, then LCS sends you an email if any server in your project goes offline without a normal shut down event. Click *Save* when you're done defining the notification rule. It then appears in a table along with other existing rules. Each has Edit and Delete Action buttons.

Notification History: Displays your web notification history in a searchable table. You can also select the date range from which to display notifications.

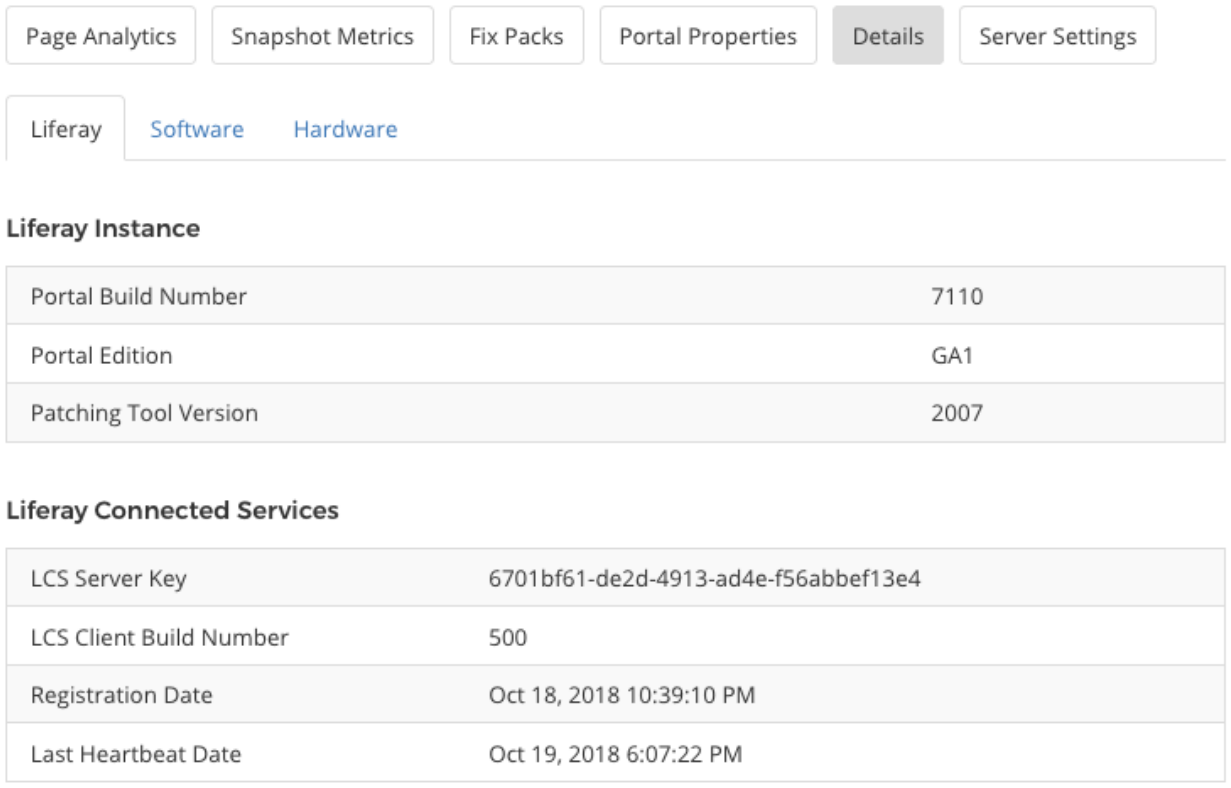


Figure 55.17: : The Details tab shows information about your server.

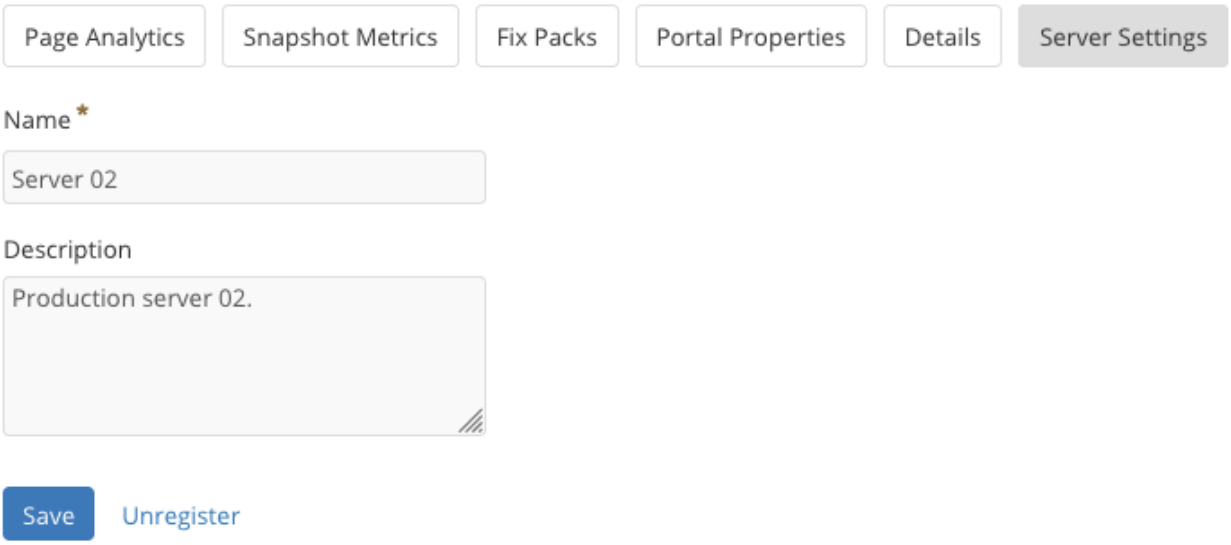


Figure 55.18: : You can use the Server Settings tab to give your server a fun name.

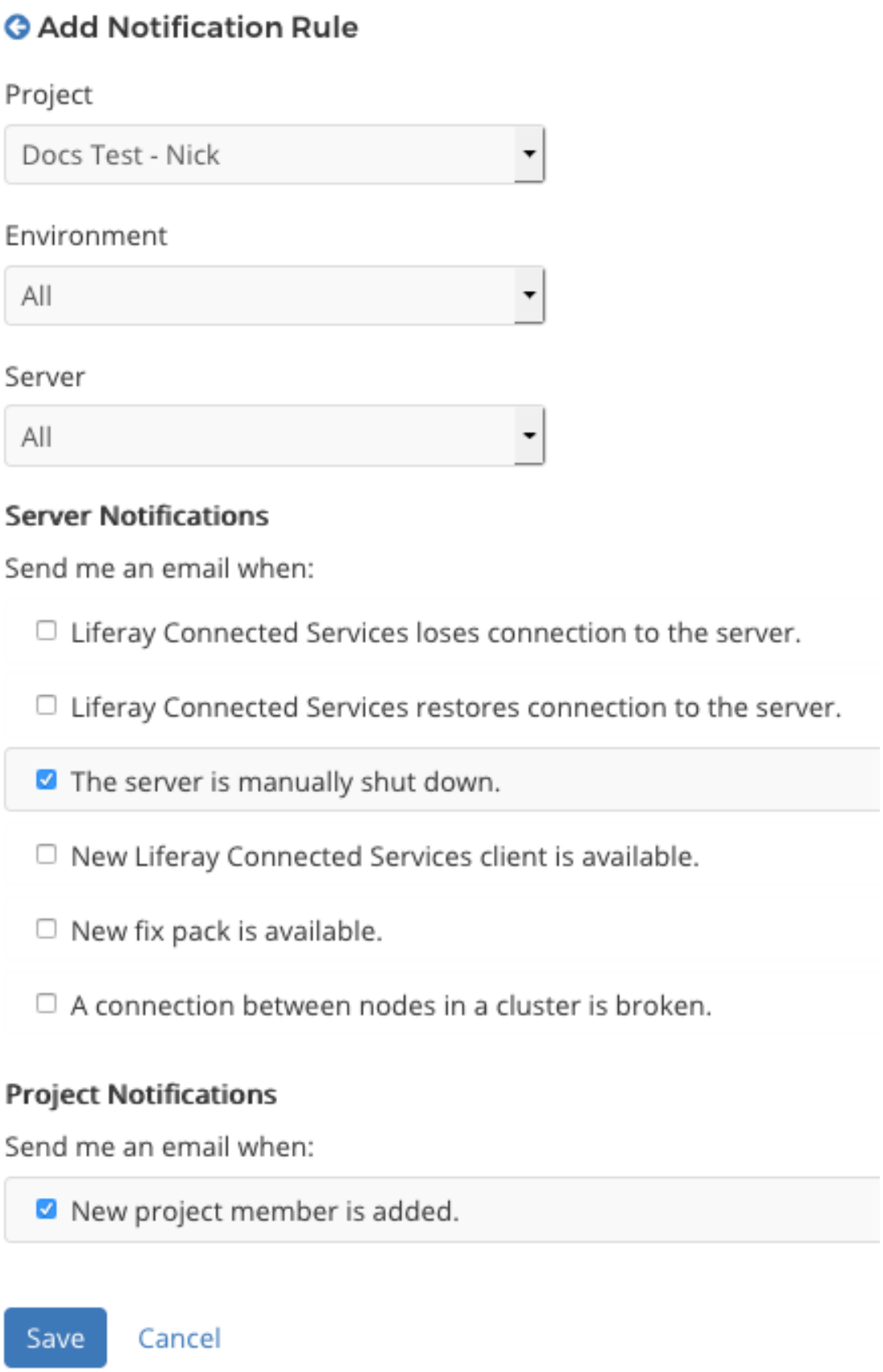


Figure 55.19: : You can add rules to determine the events that trigger notifications.

Preferences: Manage your LCS account's preferences. You can change your account's language, time zone, and default LCS project. Your default LCS project is the one shown each time you log in to LCS.

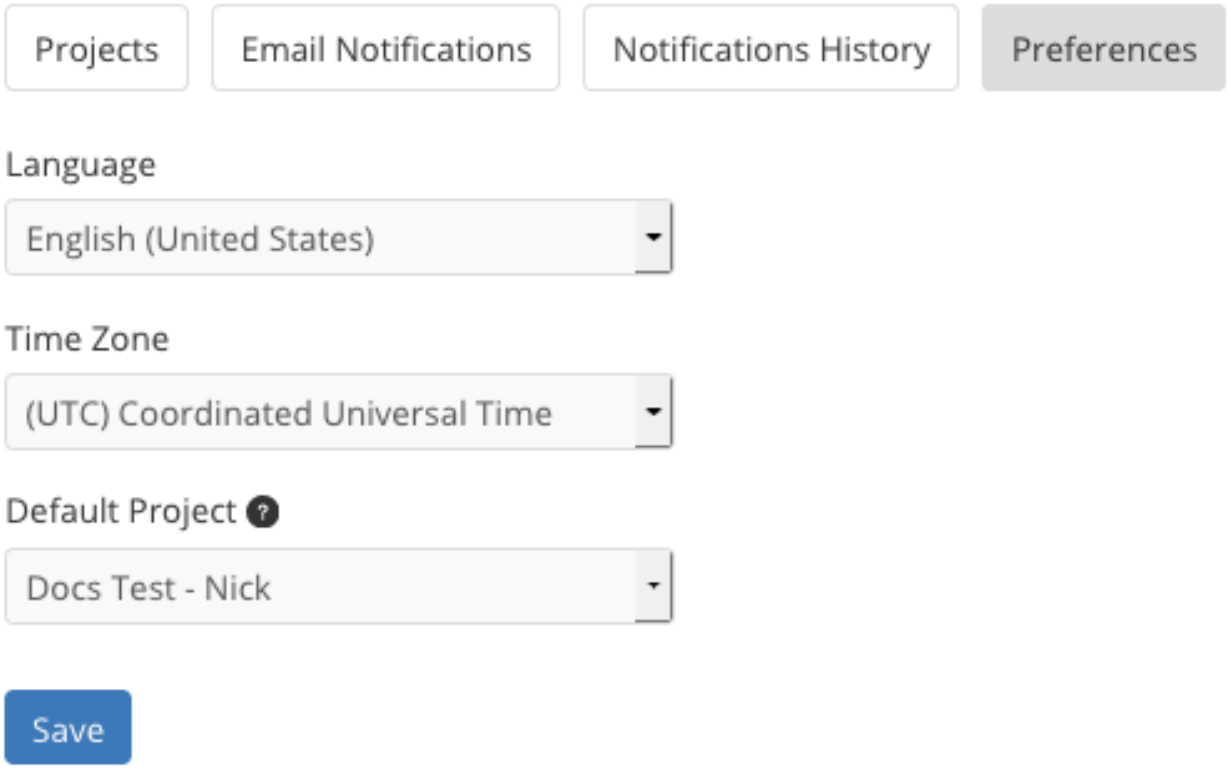


Figure 55.20: : You can change your LCS account's general preferences.

Using Web Notifications

LCS also displays web notifications under the bell icon in the Dockbar. A red badge on this icon shows your unread notification count. LCS and Liferay Support send these notifications. For example, LCS generates notifications when a server shuts down or some other event requiring your attention occurs. To mark a notification as read, click the small *x* icon next to it. To mark all notifications as read, click the *Mark All as Read* button. To mark notifications as unread again, click the *Undo* button that appears. To see your notification history, click the *Notifications History* button. You can also access your notification history by selecting *My Account* from the user menu in the Dockbar.

Managing Liferay DXP Subscriptions

LCS lets you use and view your Liferay DXP subscriptions. Recall that when you create an environment, you assign its subscription type and choose whether LCS activates servers that connect to that environment. If you use LCS for activation, registering a server in that environment consumes an activation key from the environment's subscription type. You can also view your project's available activation keys and see how they're being used.

Depending on your subscription agreement, LCS also lets you register servers via *elastic subscriptions*. Elastic subscriptions let you register an unlimited number servers. This is invaluable in auto-scaling en-

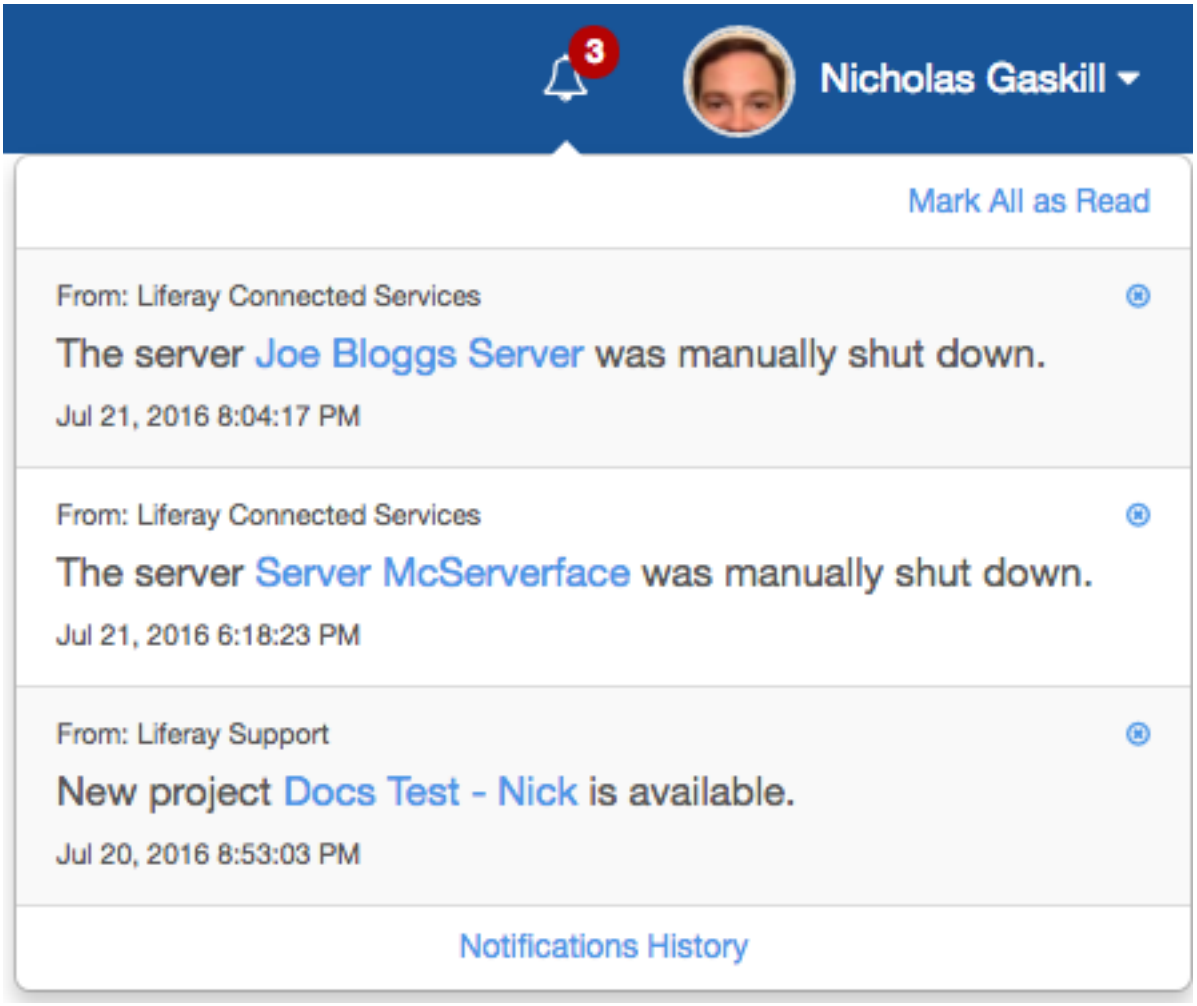


Figure 55.21: : Web notifications let you know what's happening in your LCS projects.

vironments, where servers are automatically created and destroyed in response to load. Note that to use elastic subscriptions, you must set the environment as elastic when you create it. Also note that LCS only uses elastic subscriptions for servers that exceed the number that the environment's subscription type allows. In other words, LCS uses the environment's regular subscriptions before any elastic subscriptions.

You can access these features from the *Subscriptions* tab on the upper-left of the LCS site. This tab contains two other tabs: *Details* and *Elastic Subscriptions*.

There are four tables in the *Details* tab:

1. **Subscriptions:** Shows a list of the available subscriptions in your LCS project. For each subscription, this table shows the following information:

    - Subscription Type
    - Start Date
    - Expiration Date
    - Support End Date
    - Platform
    - Product

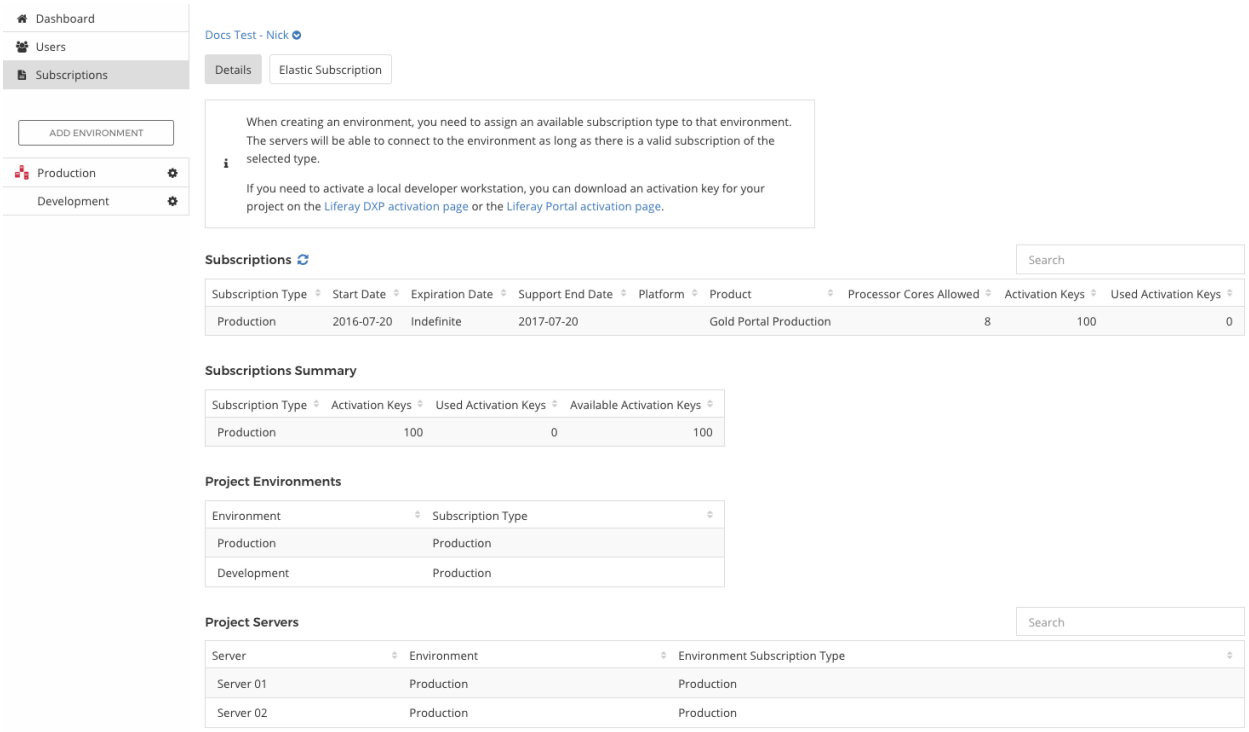


Figure 55.22: : LCS lets you view and manage your subscriptions.

- Processor Cores Allowed
- Activation Keys
- Used Activation Keys

Note that *Processor Cores Allowed* shows the number of processor cores that the subscription allows for each server.

2. **Subscriptions Summary:** Shows how your subscriptions are currently used in your project. For each subscription type, this table shows the number of activation keys allowed, used, and available.

3. **Project Environments:** Shows your project's environments and their assigned subscription types. Each environment must have a subscription type.

4. **Project Servers:** Shows the environment and subscription type for each server in your LCS project.

If any of the information in these tables is missing or incorrect, contact Liferay Support.

Note: If you don't use LCS for activating your servers, then you can register as many servers as you want in LCS.

Note: If you try to activate a server that exceeds the number of processor cores that your subscription allows per server, the activation fails and the server is locked down. A console error also indicates the server's core count. You can compare this with your subscription's processor cores allowed in LCS's Subscriptions table. To activate the server, you can either reduce the number of cores it uses (e.g., by deploying to different

server hardware, or reducing the number of virtual processors in a VM or container), or contact Liferay Sales to increase the number of processor cores that your subscription allows per server.

Decommissioning Servers

To decommission a server and free its activation key for reuse, select the server's environment on the left and then select the server. In the server's *Server Settings* tab, select *Unregister*. Also note that when you shut down a server normally, its activation key is immediately freed for reuse. If the server crashes or its shutdown is forced (e.g., kill), its activation key is freed for reuse within six minutes.

Elastic Subscriptions

Elastic subscriptions let you register an unlimited number of servers. This is crucial for auto-scaling environments where servers are created and destroyed automatically. You can view data on your elastic servers from the *Subscriptions* tab's *Elastic Subscriptions* tab.

Note: To register elastic servers in an environment, that environment must be set as elastic when it's created. For more information, see the documentation on creating environments.

Figure 55.23: : The *Elastic Subscriptions* tab shows details about your project's elastic servers.

The *Elastic Subscriptions* tab displays the number of elastic servers online and the uptime details for each. A graph shows the number of elastic servers online each day, while a table lists each elastic server's start time, end time, and duration. The total duration for servers is below the table. To download a report of the table's data, click *Download Report*. Also, you can use the *Environment* and *Month* selectors above the graph to select the environment and month to show data from, respectively. The data in both the graph and the table reflect your selections here.

Understanding Environment Tokens

To register a server in an environment, you must use that environment's token file. LCS Administrators and Environment Managers can generate and distribute this file. It contains all the information the LCS client app needs to register the server in the environment. When the server starts up, it uses the token to connect to LCS. If you use LCS for activation, the server automatically consumes an activation key from the environment's subscription upon connection. This makes it possible to activate servers automatically on startup with no interaction required.

Note: For instructions on using and managing your environment tokens, see the instructions on registering your server with LCS.

There are a few things to keep in mind when using environment tokens:

- Each environment can have only one token file. If you regenerate the token, servers using the old file are disconnected from LCS and can't reconnect until receiving the new file. If the server disconnects due to token regeneration and is running version 4.0.2 or later of the LCS client app, the server enters a 30-day grace period during which it functions normally. This gives the administrator time to use the new token file to reconnect to LCS. Servers running earlier versions of the LCS client app present users with an error page until the administrator reconnects with the new token.

- Use caution when distributing the token file, as anyone can use it to connect to your environment (and consume an activation key in your subscription if you're using LCS for activation).

- Minimal information (server name, location, etc...) is used to register a server with LCS. You can change this information from the server view in LCS at any time.

- Environment tokens connect using OAuth. Using an environment token overrides the OAuth authorization cycle. If LCS Administrators or Environment Managers have never registered servers in LCS, the first time they do so an OAuth authorization entry is created in LCS. If they've previously registered servers in LCS, their existing credentials are used when they create a token file.

- If the credentials of the LCS user who generated the token become invalid, you must generate a new token and use it to reconnect to LCS. An LCS user's credentials become invalid if the user leaves the LCS project or becomes an LCS Environment Manager or LCS Environment Viewer in a different environment.

So why bother with environment tokens at all? Besides simplifying the LCS connection process, environment tokens are valuable in auto-scaling environments where algorithms create and destroy servers automatically. In this situation, having clients that activate and configure themselves is crucial.

Note: If your auto-scaling environment creates new server nodes from a server in a system image, that server can't require human interaction during setup. When creating such an image, you must change any portal property settings that prevent automatic setup. By default, Liferay DXP's setup wizard requires human interaction. You must therefore set the `setup.wizard.enabled` property to `false` if you want your auto-scaling environment to create new nodes from the server.

55.4 Troubleshooting Your LCS Connection

Note: LCS is deprecated and will be shut down on December 31, 2021. Customers who activate LCS are advised to replace it with our latest activation key type which is suitable for virtualized environments.

For further information, please see Changes to Liferay Product Activation.

If you use LCS to activate Liferay DXP, your server must maintain its connection to LCS at all times. If this connection is interrupted, your server enters a grace period to allow for reconnection. Lengthy interruptions, however, can affect your server's uptime.

Note: You must use LCS for activation of containerized instances, cloud deployments, and instances that use Liferay Analytics Cloud and/or elastic subscriptions. Otherwise, you don't have to use LCS for activation. You can instead request an XML activation key from Liferay Support.

The following sections in this document provide some background information and help you troubleshoot problems with your server's LCS connection:

- **LCS Grace Periods:** Describes how grace periods work in LCS. You should read this section before attempting any troubleshooting steps.
- **Troubleshooting:** Presents troubleshooting steps for specific problems.
- **Increasing Log Levels:** If you contact Liferay Support, you'll be asked to increase your server's log levels and then provide your log files. This section shows you how to do this.

Note: The odds of LCS being unavailable are low. LCS is deployed on a global cloud infrastructure set up for automatic failure recovery. Notifications also let the LCS team react quickly to any downtime. During LCS updates and new version releases, however, LCS is unavailable for a few minutes while changes are applied.

LCS Grace Periods

There are 2 grace period types in LCS:

1. **Connection Grace Period:** Occurs when your activated LCS connection is interrupted. This gives you time to re-establish the connection.

2. **Subscription Grace Period:** Occurs when your subscription is about to expire. This gives you time to renew the subscription.

Note: These grace periods only apply to servers previously connected and activated in LCS. If the subscription check or connection fails when a server attempts to connect to LCS for the first time, that server doesn't enter a grace period. It's therefore important to verify that an active subscription is available before connecting a new server to LCS. To do this, check the Subscriptions tab in LCS.

Connection Grace Period

If your server's LCS connection is interrupted, the server continues to run and enters a grace period that lasts for up to 30 days to allow for reconnection. During this grace period, Liferay DXP displays a warning message to administrators. Upon seeing this message, administrators should contact Liferay Support and follow the troubleshooting steps below. LCS automatically restores your server's activation upon reconnection (you shouldn't need to restart the server). If for some reason the connection can't be restored, Liferay Support will provide an alternative way to activate your server.

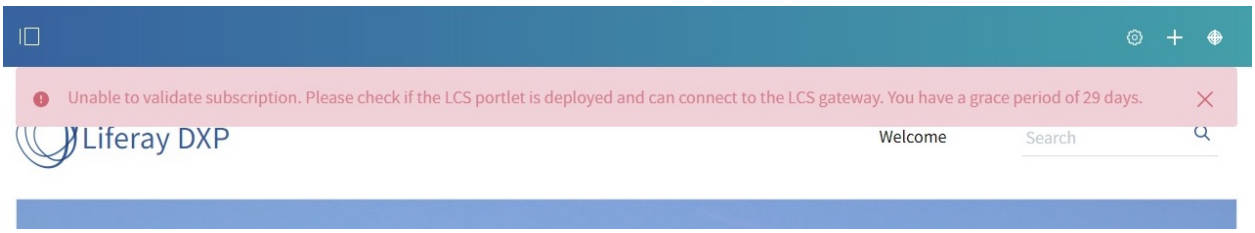


Figure 55.24: A warning message is displayed to administrators if the server can't connect to LCS to validate the subscription.

While disconnected from LCS, the LCS client app continually attempts to reconnect. If reconnection continues to fail, ensure that your server can access `lcs.liferay.com` and `lcs-gateway.liferay.com`. If the LCS client app stops attempting to reconnect, there will be no activity in the logs. In this case, you can force reconnection by redeploying the app. Follow these steps to do so:

1. In your server's Liferay Home folder (usually the parent folder of the application server's folder), remove this file:

   ```
   osgi/marketplace/Liferay Connected Services Client.lpkg
   ```

2. Place `Liferay Connected Services Client.lpkg` in `[Liferay Home]/deploy`. If you connect to LCS through a proxy, and configured this inside the LCS client app, make sure the app you deploy is also configured to do so.

You should also ensure that you've enabled email notifications in LCS for server disconnection events. To do this, you must create a notification rule that sends an email whenever the server shuts down unexpectedly. The documentation on managing Your account explains how to do this.

LCS's grace period behavior has been implemented in several patches. If you're not running these patches, your server's grace period may be different. The following table lists each patch and how it changes the grace period. Note that each hotfix includes the preceding hotfixes, and each fix pack includes the preceding fix packs.

Fix Pack	Hotfix Built After	Before	After
40	05 March 2018	The grace period lasts 7 days.	The grace period lasts 30 days.

Fix Pack	Hotfix Built After	Before	After
33	17 Nov 2017	The grace period is only invoked for lost network connections.	The grace period is also invoked for failed subscription validation, if the server was previously registered.
32	20 Oct 2017	The grace period warning is displayed as soon as the LCS connection is lost.	The grace period warning is displayed only after the connection has been out for 1 hour. This prevents false alarms for temporary network problems.

Subscription Grace Period

At least 90 days before the subscription expires, Liferay will reach out to begin the renewal process. 30 days before expiration, Liferay Support sends warning messages through the Help Center, the LCS site, and the Customer Portal. After the expiration date, your servers may be placed in an additional grace period, which is communicated through the same support channels. If the renewal isn't completed during this grace period, then the subscription becomes inactive and the Liferay DXP instance enters the 30-day grace period. As soon as the renewal is processed, the instance activates and any error or warning messages disappear within 24 hours. Note that by using XML activation keys (provided by Liferay Support upon request), you can continue to use your Liferay DXP instances even after a subscription has expired.

Troubleshooting

If you encounter issues with LCS, the Liferay Support team is here to help. If you need support, open a Help Center ticket. You can begin troubleshooting the following scenarios, which the Liferay Support team can also assist you with.

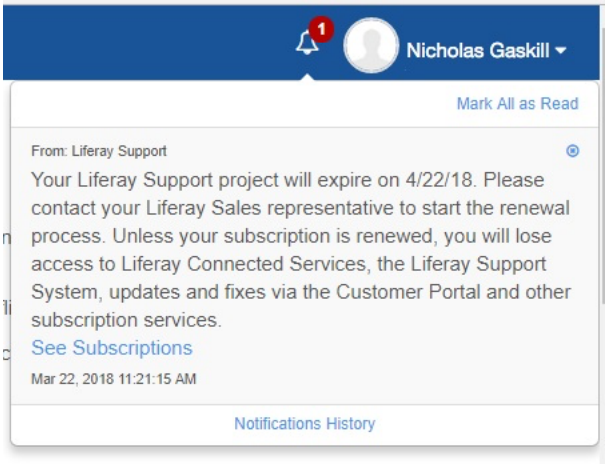


Figure 55.25: LCS sends you a notification prior to the expiration of your subscription.

Note: Before troubleshooting specific issues or contacting Liferay Support, make sure that you've followed the LCS preconfiguration and registration steps correctly.

Server Can't Reach LCS

If your server can't reach LCS, verify that you can access the public sites required by LCS:

- `lcs.liferay.com` should be viewable in a browser.
- `lcs-gateway.liferay.com` should respond on port 443:

```
curl -vk -I "https://lcs-gateway.liferay.com"
telnet lcs-gateway.liferay.com 443
```

Subscription Issues

For issues related to your subscription, first review the documentation on managing your subscription. Subscription errors usually involve one of these problems:

- Your server can reach LCS, but can't locate a subscription.
- Your server can reach LCS and locate a subscription, but activating your server would exceed the subscription's number of activation keys or cores.

In either case, you must verify that a subscription is available and that you're not exceeding its number of activation keys or cores. You can find this information on the LCS site's Subscriptions page, as described in the documentation on managing subscriptions. If the environment in which you're trying to activate a server isn't assigned the subscription you want to use, then you must create a new environment and assign it the correct subscription. Once assigned, you can't change an environment's subscription. Follow the initial registration steps for instructions on creating a new environment and activating a new server.

Note: When shutting down servers, you must ensure that the LCS site receives the server shutdown commands. Otherwise, LCS may not release that server's activation key for reuse and attempts to activate additional servers may exceed the subscription's number of activation keys. There's a higher likelihood of

this happening in rolling deployments and/or when using containers. For more information, see the KB article on properly unregistering subscriptions.

Invalid Token

If the token is invalid, first review the documentation on environment tokens. The following table lists causes and solutions for invalid tokens.

Cause	Solution
The LCS user who generated the token no longer has permissions. This happens when the user leaves the LCS project or becomes an LCS Environment Manager or LCS Environment Viewer in a different environment.	Regenerate the token.
The token's file name is changed after download.	Download the token again from LCS.
The token is regenerated.	Use the regenerated token.

Increasing Log Levels

If you contact Liferay Support, you're asked to increase your server's log levels and then provide your log files. You can find these log files in `[Liferay Home]/logs` (Liferay Home is usually the parent folder of the application server's folder). There are 2 types of log files in this folder:

1. **Liferay log files:** The files `liferay.[date].log` and `liferay.[date].xml` are the logs for your Liferay DXP installation. Note that LOG and XML files for the same date contain the same information—the only difference is the file format.

2. **LCS log files:** The `lcs-portlet-[date].log` files are the LCS client app's logs. Note that if there's only a single LCS log file, it may appear without a date as `lcs-portlet.log`. When you increase the log levels as described in the following sections, the additional log messages are written to these LCS log files.

There are 2 ways to increase the log levels:

1. **In your Liferay DXP instance's Control Panel:** This is a temporary configuration that resets upon shutting down the server. Note that if the server isn't activated, you can't access the Control Panel. In that case, Liferay Support can provide an XML activation key.

2. **In a Log4j configuration:** This is a permanent configuration that persists through server shutdown and restart.

The following sections cover both options.

Control Panel

Follow these steps to increase the log levels via the Control Panel:

1. Navigate to *Control Panel → Configuration → Server Administration*.
2. Click the *Log Levels* tab.
3. Search for "lcs".
4. Change the log level for each matching entry to DEBUG.
5. While in the Control Panel, you should also navigate to *Configuration → Liferay Connected Services* and take a screenshot of what you see there. This is useful to Liferay Support.

Log4j

Follow these steps to increase the log levels via Log4j:

1. Download the latest LCS client as instructed in the LCS preconfiguration article. The app downloads as `Liferay Connected Services Client.lpkg`. If you don't want to download the latest client, you can use the one already installed in your server: it's in `[Liferay Home]/osgi/marketplace` (just make sure to shut down your server before following the rest of the steps in this section). Recall that the Liferay Home folder is usually the parent folder of the application server's folder.
2. Expand the LPKG file, then expand the `lcs-portlet-[version].war` file inside it.
3. Inside the `WAR` file, replace the contents of `WEB-INF\classes\META-INF\portal-log4j.xml` with the following configuration:

```
<?xml version="1.0"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
	<appender class="org.apache.log4j.rolling.RollingFileAppender" name="RollingFileAppender">
		<rollingPolicy class="org.apache.log4j.rolling.TimeBasedRollingPolicy">
			<param name="ActiveFileName" value="@example.home@/logs/lcs-portlet.log" />
			<param name="FileNamePattern" value="@example.home@/logs/lcs-portlet.%d{yyyy-MM-dd}.log.zip" />
		</rollingPolicy>

		<layout class="org.apache.log4j.EnhancedPatternLayout">
			<param name="ConversionPattern" value="%d{yyyy/MM/dd HH\:mm\:ss} %-5p [%t][%c{1}:%L] %m%n" />
		</layout>
	</appender>

	<category name="com.liferay.lcs.task.scheduler">
		<priority value="ALL" />
	</category>

	<logger additivity="false" name="com.liferay.lcs">
		<level value="ALL" />
		<appender-ref ref="RollingFileAppender" />
	</logger>
</log4j:configuration>
```

4. Save the file and repackage the WAR and LPKG (make sure not to change the names of these files).
5. Make sure your server is shut down.

6. In your installation's Liferay Home folder, delete the existing LCS client app:

   ```
   osgi/marketplace/Liferay Connected Services Client.lpkg
   ```

7. Place the `Liferay Connected Services Client.lpkg` that you repackaged in step 4 in `osgi/marketplace`.

8. Start your server.

If you need assistance with the issues in this guide, or any other issues with LCS, contact Liferay Support.

55.5 Building Clustering for Liferay Portal

Clustering was added back to the CE product. To use it, upgrade to a current Liferay release.

55.6 Configuration Reference

The Configuration Reference page gives you shortcuts to Liferay Portal configuration reference documentation. From here, you can easily check Liferay definitions and configuration defaults as you customize your portal.

portal.properties(Opens New Window)
Describes the configuration defaults for Liferay Portal.

system.properties(Opens New Window)
Describes the system configuration defaults for Liferay Portal.

55.7 Database Templates

Below are templates for configuring the various databases that Liferay DXP supports with Liferay's built-in data source.

MariaDB

```
jdbc.default.driverClassName=org.mariadb.jdbc.Driver
jdbc.default.url=jdbc:mariadb://localhost/lportal?useUnicode=true&characterEncoding=UTF-8&useFastDateParsing=false
jdbc.default.username=
jdbc.default.password=
```

MySQL

```
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.url=jdbc:mysql://localhost/lportal?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDateParsin
jdbc.default.username=
jdbc.default.password=
```

PostgreSQL

```
jdbc.default.driverClassName=org.postgresql.Driver
jdbc.default.url=jdbc:postgresql://localhost:5432/lportal
jdbc.default.username=sa
jdbc.default.password=
```

55.8 Elasticsearch Settings

Elasticsearch is the default search engine for 7.0. The *Liferay Foundation* suite includes an adapter for Elasticsearch called *Liferay Portal Search Elasticsearch*. The adapter is configurable through System Settings or an OSGi configuration file named `com.liferay.portal.search.elasticsearch.configuration.ElasticsearchConfiguration.cfg` and deployed to `[Liferay_Home]/osgi/configs`.

Elasticsearch 6: The Liferay Connector to Elasticsearch 6 contains all the settings of the default adapter, along with a few new ones. See the end of this article to learn about the settings unique to the Elasticsearch 6 connector.

The list below is all the configuration settings for Liferay's default Elasticsearch 2.x adapter, in the order they appear in the System Settings application:

`clusterName=LiferayElasticsearchCluster` A String value that sets the name of the cluster to integrate with. This name should match the remote cluster when Operation Mode is set to remote. (See also: remote operation mode)

`operationMode=EMBEDDED` There are two operation modes you can choose from: EMBEDDED or REMOTE. Set to REMOTE to connect to a remote standalone Elasticsearch cluster. Set to EMBEDDED to start Liferay with an internal Elasticsearch instance. Embedded operation mode is unsupported for production environments.

`indexNamePrefix=liferay-` Set a String value to use as the prefix for the search index name. The default value should not be changed under normal conditions. If you change it, you must also perform a *reindex all* operation for the portal and then manually delete the old index using the Elasticsearch administration console.

`bootstrapMlockAll=false` A boolean setting that, when set to true, tries to lock the process address space into RAM, preventing any Elasticsearch memory from being swapped out (see here) for more information)

`logExceptionsOnly=true` A boolean setting that, when set to true, only logs exceptions from Elasticsearch, and does not rethrow them.

`retryOnConflict=5` Set an int value for the number of retries to attempt if a version conflict occurs because the document was updated between getting it and updating it (see here for more information).

`discoveryZenPingUnicastHostsPort=9300-9400` Set a String value for the range of ports to use when building the value for discovery.zen.ping.unicast.hosts. Multiple Elasticsearch nodes on a range of ports can act as gossip routers at the same computer (see here for more information).

`networkHost=` Set this String value to instruct the node to bind to this hostname or IP address and publish (advertise) this host to other nodes in the cluster. This is a shortcut which sets the bind host and the publish host at the same time (see here for more information).

`networkBindHost=` Set the String value of the network interface(s) a node should bind to in order to listen for incoming requests (see here for more information).

`networkPublishHost=` Set the String value of a single interface that the node advertises to other nodes in the cluster, so that those nodes can connect to it (see here for more information).

`transportTcpPort=` Set the String value for the port to bind for communication between nodes. Accepts a single value or a range (see here for more information).

`transportAddresses=localhost:9300` Set the String values for the addresses of the remote Elasticsearch nodes to connect to. This value is required when Operation Mode is set to remote (see here for more information). Specify as many or few nodes as you see fit.

`clientTransportSniff=true` Set this booleant to true to enable cluster sniffing and dynamically discover available data nodes in the cluster (see here for more information).

clientTransportIgnoreClusterName=false Set this boolean to true to ignore cluster name validation of connected nodes (see here for more information).

clientTransportNodesSamplerInterval=5s Set this String value to instruct the client node on how often to sample / ping the nodes listed and connected (see here for more information).

httpEnabled=true Set this boolean to false to disable the http layer entirely on nodes which are not meant to serve REST requests directly (see here for more information).

httpCORSEnabled=true Set this boolean to false to disable cross-origin resource sharing, i.e. whether a browser on another origin can do requests to Elasticsearch. If disabled, web front end tools like elasticsearch-head may be unable to connect (see here for more information).

httpCORSAllowOrigin=/https?:\\/\\/localhost(:[0-9]+)?/ Set the String origins to allow when HTTP CORS is enabled (see here for more information).

httpCORSConfigurations= Set the String values for custom settings for HTTP CORS, in YML format (elasticsearch.yml) (see here for more information).

additionalConfigurations= Set the String values for custom settings for embedded Elasticsearch, in YML format. See: Adding Settings to the Liferay Elasticsearch Adapter

additionalIndexConfigurations= Set the String values for custom settings for the Liferay index, in JSON or YML format (refer to the Elasticsearch Create Index API for more information). See: Adding Settings to the Liferay Elasticsearch Adapter

additionalTypeMappings= Set the String values for custom mappings for the `LiferayDocumentType`, in JSON format (refer to the Elasticsearch Put Mapping API for more information) See: Adding Settings to the Liferay Elasticsearch Adapter

The following settings are only available in the Elasticsearch 6 adapter:

indexNumberOfReplicas= Set the String value for how many replica shards each primary shard has.

indexNumberOfShards= Set the String value for the number of primary shards an index has.

clientTransportPingTimeout= Set the String value for how long to wait for a node's ping response.

Configurations only Affecting the Embedded Elasticsearch Server

These settings (defined above) are only meant to use while configuring the embedded Elasticsearch server. Configuring these will elicit no effect on remote Elasticsearch installations:

`bootstrapMlockAll` `discoveryZenPingUnicastHostsPort` `networkHost` `networkBindHost` `networkPublishHost` `transportTcpPort` `httpEnabled` `httpCORSEnabled` `httpCORSAllowOrigin` `httpCORSConfigurations`

You can easily configure these settings in the System Setting application, or as mentioned above, you can specify them in a deployable OSGi `.config` file.

55.9 Form Field Types

A form without fields is no form at all. To meet your form-building needs, Liferay Forms provides useful and highly configurable field types. To see examples of the form fields described here, visit the Collecting Information from Users section of the User Guide.

Form Text This is static text on the form. Users do not enter data into Form Text fields. The form creator enters text that form users see displayed on the form. This is useful for longer instructions.

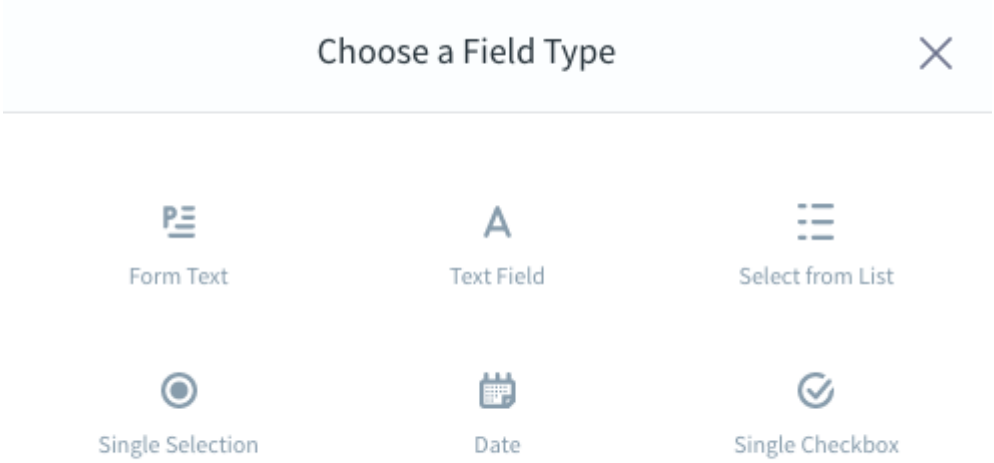


Figure 55.26: Out of the box form field types.

Figure 55.27: A form text field.

Text Field Users enter text into these fields. For example, a Full Name field is a text field. By default, a text field keeps all input on a single line of text. To accommodate longer responses, choose the multi-line setting when configuring the text field as in this example. Place limits on the text users can enter (numbers from 1-10, or email addresses, for example) by using the Text field's validation options (as in this example).

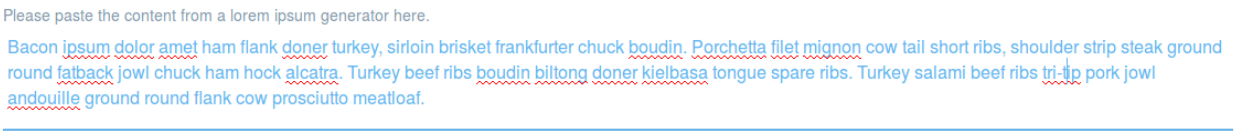


Figure 55.28: A multiline text form.

Select from List Users select one (or more, if configured to allow it) options from a list of choices. Choices are entered manually or automatically populated by a data provider. For example, a Country of Residence field can be a select from list field populated by a Countries of the World data provider.

Single Selection Using a radio button, users select one option from a list of options displayed on the form.

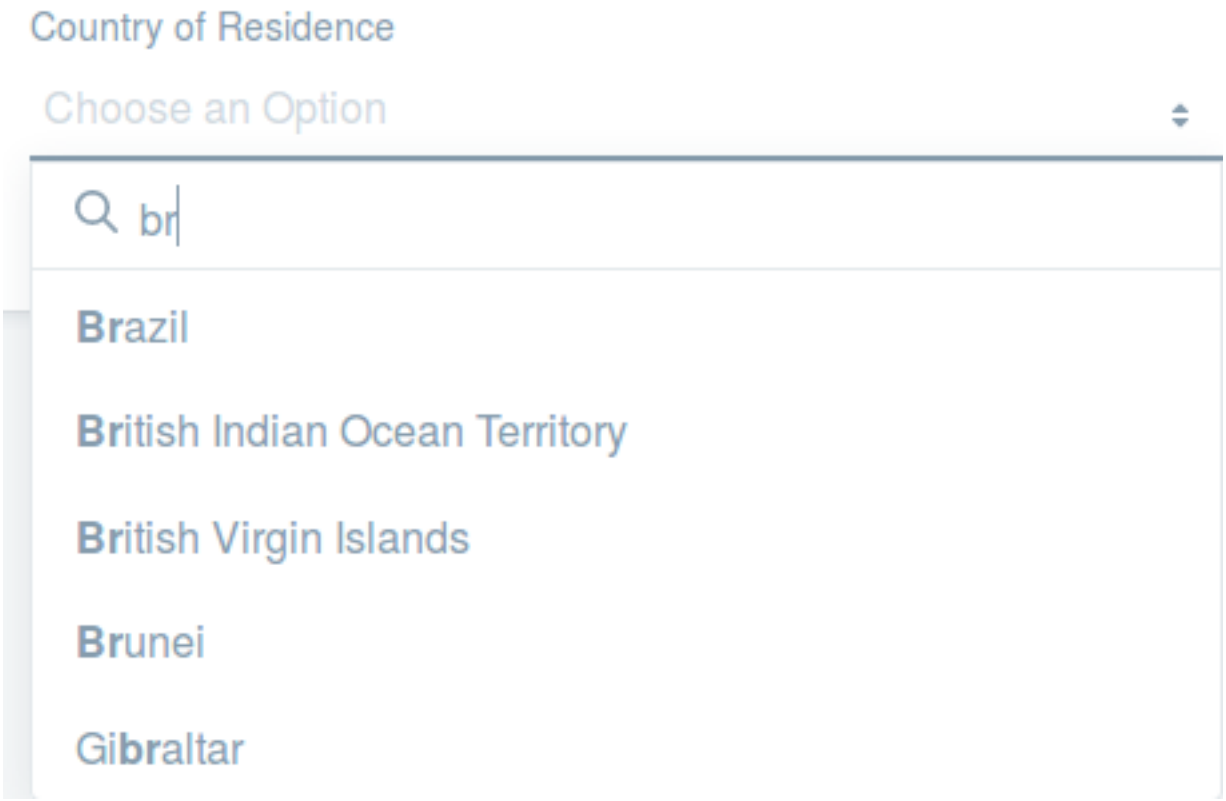


Figure 55.29: A select from list field.

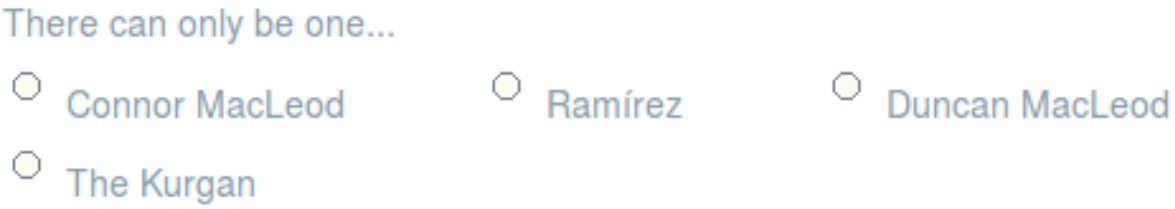


Figure 55.30: A single selection field.

Date Users select a date using a date picker. For example, a Birth Date field uses the Date field type.

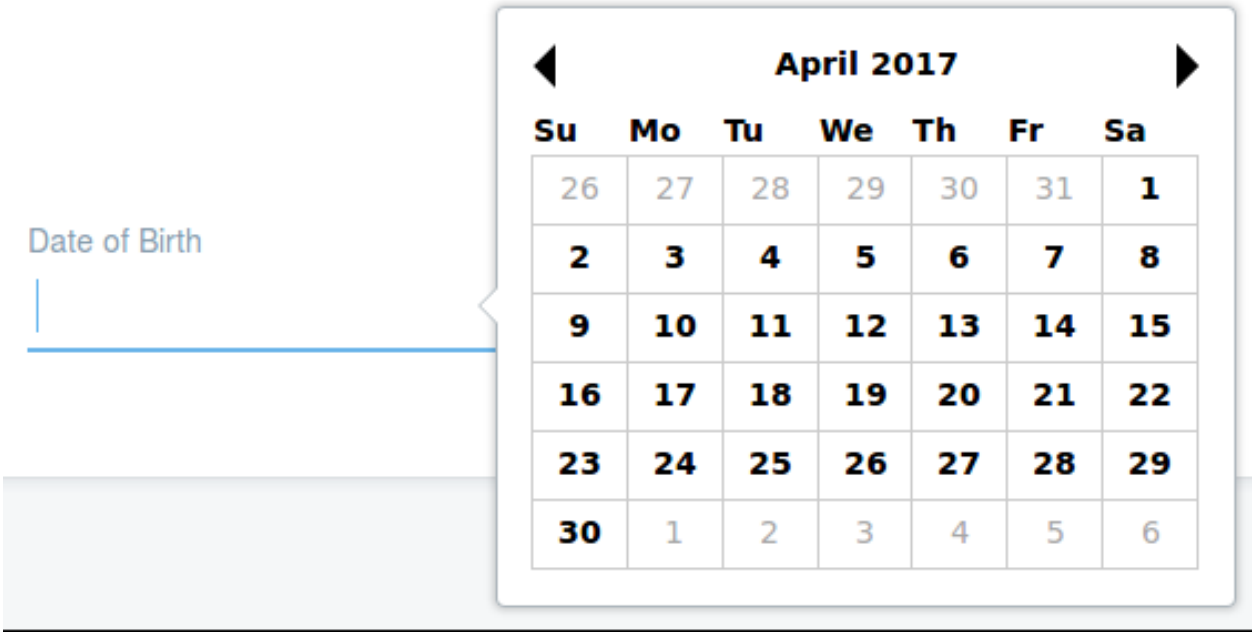


Figure 55.31: A date field.

Multiple Selection/Single Checkbox Users select one or more options from check boxes (or switchers, if configured). The Single Checkbox field is only available to Digital Enterprise subscribers, but its functionality can be obtained with the Multiple Selection field. The Single Checkbox field will be removed in the next version of Liferay DXP Digital Enterprise.

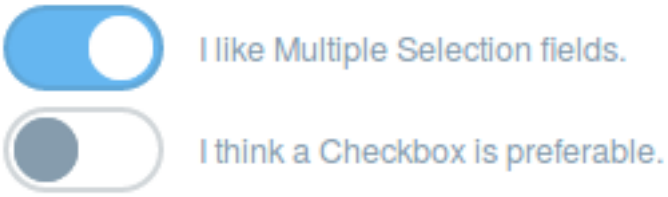


Figure 55.32: A multiple selection field using a switcher.

55.10 Deployment Reference

Here you'll find definitions, default settings, templates, and more. Here are some of the topics:

Database Templates: URL patterns for connecting to supported databases.
Elasticsearch Settings: Configuration options for Liferay DXP's search engine.
Form Field Types: A list of all the possible variations of form fields.

55.11 Knowledge Base Configuration

These are the settings to configure the Knowledge Base, including the migration of properties from Liferay Portal 6.2.

This reference doc has two sections:

- Knowledge Base System Settings
- Overriding Settings

Knowledge Base System Settings

Configuration changes made in System Settings set the default configuration for the corresponding app across all sites. Knowledge Base apps in your site can override these default values through their *Configuration* menu.

Knowledge Base Article Configuration

- **Resource Prim key:** Sets the primary key that identifies a Knowledge Base article or Knowledge Base folder (depending on the resource class name ID). The default value is `0`.
- **Resource class name id:** Sets the class name of either the Knowledge Base article or Knowledge Base folder. This value needs to match the value of the `Resource Prim key`. If the class name is a Knowledge Base article, the resource prim key needs to be the primary key of a Knowledge Base article. If the class name belongs to a Knowledge Base folder the resource prim key needs to belong to a Knowledge Base folder. The default value is `0`.
- **Enable KBArticle description:** Sets whether to display the description field in the edit view of the application. This is disabled by default.
- **Enable KBArticle ratings:** Sets whether to allow users to add ratings to the Knowledge Base Article. This is enabled by default.
- **Show KBArticle asset entries:** Sets whether to show the asset entries for the Knowledge Base Article. This is displayed by default.
- **Show KBArticle attachments:** Sets whether to show the attachments for the Knowledge Base article. This is displayed by default.
- **Show KBArticle asset links:** Sets whether to show the asset links for the Knowledge Base Article. This is displayed by default.
- **Enable KBArticle view count increment:** Sets whether to add a view counter to the Knowledge Base article. This is disabled by default.
- **Enable KBArticle subscriptions:** Sets whether to allow user to subscribe to the Knowledge Base article. This is enabled by default.
- **Enable KBArticle history:** Sets whether to allow users to view the history for the Knowledge Base article. This is enabled by default.
- **Enable KBArticle print:** Sets whether to give users the option to print the Knowledge Base article. This is enabled by default.

- **Enable social bookmarks:** Sets whether to display social bookmarks when viewing a Knowledge Base article. This is disabled by default.
- **Social bookmarks display style:** Sets the display style of the social bookmarks. Possible values are `menu`, `simple`, `vertical`, or `horizontal`. The default value is `menu`.
- **Social bookmarks display position:** Sets the position to display social bookmarks. Possible values are `top` or `bottom`. The default value is `bottom`.
- **Social bookmarks types:** A comma-separated list of the possible bookmark types (`twitter,facebook,plusone` for example). Override with a `portal-ext.properties` file. The default value is `${server-property://com.liferay.portal/social.bookmark.types}`.

Knowledge Base Display Configuration

- **Resource Prim key:** Sets the primary key that identifies a Knowledge Base article or Knowledge Base folder (depending on the resource class name ID). The default value is `0`.
- **Resource class name id:** Sets the class name of either the Knowledge Base article or Knowledge Base folder. This value needs to match the value of the `Resource Prim key`. If the class name is a Knowledge Base article, the resource prim key needs to be the prim key of a Knowledge Base article. If the class name belongs to a Knowledge Base folder the resource prim key needs to belong to a Knowledge Base folder. The default value is `0`.
- **Enable KBArticle description:** Sets whether to display the description field in the edit view of the application. This is disabled by default.
- **Enable KBArticle ratings:** Sets whether to allow users to add ratings to the Knowledge Base Article. This is enabled by default.
- **Show KBArticle asset entries:** Sets whether to show the asset entries for the Knowledge Base Article. This is displayed by default.
- **Show KBArticle attachments:** Sets whether to show the attachments for the Knowledge Base article. This is displayed by default.
- **Show KBArticle asset links:** Sets whether to show the asset links for the Knowledge Base Article. This is displayed by default.
- **Enable KBArticle view count increment:** Sets whether to add a view counter to the Knowledge Base article. This is enabled by default.
- **Enable KBArticle subscriptions:** Sets whether to allow user to subscribe to the Knowledge Base article. This is enabled by default.
- **Enable KBArticle history:** Sets whether to allow users to view the history for the Knowledge Base article. This is enabled by default.
- **Enable KBArticle print:** Sets whether users can print the Knowledge Base article. This is enabled by default.
- **Enable social bookmarks:** Sets whether to display social bookmarks when viewing a Knowledge Base article. This is disabled by default.

- **Social bookmarks display style:** Sets the display style of the social bookmarks. Possible values are `menu`, `simple`, `vertical`, or `horizontal`. The default value is `menu`.
- **Social bookmarks display position:** Sets the position to display social bookmarks. Possible values are `top` or `bottom`. The default value is `bottom`.
- **Social bookmarks types:** A comma-separated list of the possible bookmark types (`twitter,facebook,plusone` for example). Override with a `portal-ext.properties` file. The default value is `${server-property://com.liferay.portal/social.bookmark.types}`.
- **Content root prefix:** Sets the default value for the content root prefix.

Knowledge Base Service Configuration

The list below describes what each configuration option does in the System Settings UI.

- **Get editor name:** Sets the default editor. Possible values are: `alloyeditor`, `ckeditor`, `simple`, `tinymce`, and `tinymce_simple`. The default value is `alloyeditor`.
- **Article increment priority enabled:** Sets whether to increment article priority by `1.0`. This is enabled by default.
- **Markdown importer article extensions:** Sets the supported article file extensions for the Markdown importer. Add a new field for each file extension you wish to use. Note that the abbreviated and full file extensions will need to be specified if you wish to use both (`.markdown` and `.md` for example). The default values are `.markdown` and `.md`.
- **Markdown importer article intro:** Sets the article parent file suffix for the Markdown importer. The default value is `intro.markdown`. Any Markdown files named `[name]-intro.markdown` would be considered the parent article in the folder, with the default setting. The default value is `intro.markdown`.
- **Markdown importer image file extensions:** Sets the supported image file extensions for the Markdown importer. Add a new field for each file extension you wish to use. Note that the abbreviated and full file extensions will need to be specified if you wish to use both (`.jpeg` and `jpg` for example). The default values are `.bmp`, `.gif`, `.jpeg`, `.jpg`, and `.png`.
- **Markdown importer image folder:** Sets the image folder path the Markdown importer looks for in the ZIP file. The default value is `/images`.
- **Source URLEnabled:** Sets whether users can access the source location of importable Markdown files. If enabled, a button appears above each article that links them to the source location for the article. This is disabled by default.
- **Source URLEdit message key:** Sets the language key to use for the source URL button label. The default value is `edit-on-github`.
- **Email from name:** Sets the Sender's name for automated emails. Override with a `portal-ext.properties` file. The default value is `${server-property://com.liferay.portal/admin.email.from.name}`.
- **Email from address:** Sets the Sender's email address for automated emails. Override with a `portal-ext.properties` file. The default value is `${server-property://com.liferay.portal/admin.email.from.address}`.

- **Email KBArticle added enabled:** Sets whether to send an automated email when a Knowledge Base article is added. This is enabled by default.
- **Email KBArticle added subject:** Sets the subject template for the Knowledge Base article added email. The default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_article_added_subject.tmpl}`.
- **Email KBArticle added body:** Sets the body template for the Knowledge Base article added email. The default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_article_added_body.tmpl}`.
- **Email KBArticle updated enabled:** Sets whether to send an automated email when a Knowledge Base article is updated. This is enabled by default.
- **Email KBArticle updated subject:** Specifies the email template for the subject of the updated Knowledge Base article email. This is the default value `${resource:com/liferay/knowledge/base/dependencies/email_kb_artic`
- **Email KBArticle updated body:** Specifies the email template for the body of the updated Knowledge Base article email. The default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_article_update`
- **Email KBArticle suggestion in progress enabled:** Sets whether to send an automated email when a Knowledge Base article suggestion is moved to in progress. This is enabled by default.
- **Email KBArticle suggestion in progress subject:** Specifies the email template for the subject of the in progress email. The default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_suggestion_in_pr`
- **Email KBArticle suggestion in progress body:** Specifies the email template for the body of the in progress email. The default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_suggestion_in_pr`
- **Email KBArticle suggestion received enabled:** Sets whether to send an automated email when a Knowledge Base article suggestion is received. This is enabled by default.
- **Email KBArticle suggestion received subject:** Specifies the email template for the subject of the suggestion received email. the default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_suggestion_re`
- **Email KBArticle suggestion received body:** Specifies the email template for the body of the suggestion received email. The default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_suggestion_receiv`
- **Email KBArticle suggestion resolved enabled:** Sets whether to send an automated email when a Knowledge Base article suggestion is resolved. This is enabled by default.
- **Email KBArticle suggestion resolved subject:** Specifies the email template for the subject of the suggestion resolved email. The default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_suggestion_re`
- **Email KBArticle suggestion resolved body:** Specifies the email template for the body of the suggestion resolved email. The default value is `${resource:com/liferay/knowledge/base/dependencies/email_kb_suggestion_resolv`
- **Enable RSS:** Sets whether to enable the RSS feed. This is enabled by default.
- **Rss delta:** Sets the pagination for RSS entries. The default value is `20`.
- **Rss display style:** Sets the display style of the RSS. Possible values are `abstract`, `full-content`, or `title`. The default value is `full-content`.
- **Rss format:** Sets the web feed language to use for your RSS feed. Possible values are `atom10` (Atom 1.0 the default), `rss10`(RSS 1.0), or `rss20`(RSS 2.0). The default value is `atom10`.
- **Rss feed type:** Sets the default feed type from the feed types defined in the `rss.feed.types` portal property. The default value is `${server-property://com.liferay.portal/rss.feed.type.default}`.

Knowledge Base Search Configuration

- **Show KBArticle author column:** Sets whether to display the author column in the search results of the application. This is enabled by default.
- **Show KBArticle create date column:** Sets whether to display the create date column in the search results of the application. This is enabled by default.
- **Show KBArticle modified date column:** Sets whether to display the modified date column in the search results of the application. This is displayed by default.
- **Show KBArticle views column:** Sets whether to display the views column in the search results of the application. This is displayed by default.
- **Enable KBArticle description:** Sets whether to display the description field in the edit view of the application. This is disabled by default.
- **Enable KBArticle ratings:** Sets whether to allow users to add ratings to the Knowledge Base Article. This is enabled by default.
- **Show KBArticle asset entries:** Sets whether to show the asset entries for the Knowledge Base Article. This is displayed by default.
- **Show KBArticle attachments:** Sets whether to show the attachments for the Knowledge Base article. This is displayed by default.
- **Show KBArticle asset links:** Sets whether to show the asset links for the Knowledge Base Article. This is displayed by default.
- **Enable KBArticle view count increment:** Sets whether to add a view counter to the Knowledge Base article. This is enabled by default. This is enabled by default.
- **Enable KBArticle subscriptions:** Sets whether to allow user to subscribe to the Knowledge Base article. This is enabled by default.
- **Enable KBArticle history:** Sets whether to allow users to view the history for the Knowledge Base article. This is enabled by default.
- **Enable KBArticle print:** Sets whether to give users the option to print the Knowledge Base article. This is enabled by default.
- **Enable social bookmarks:** Sets whether to display social bookmarks when viewing a Knowledge Base article. This is disabled by default.
- **Social bookmarks display style:** Sets the display style of the social bookmarks. Possible values are `menu`, `simple`, `vertical`, or `horizontal`. The default value is `menu`.
- **Social bookmarks display position:** Sets the position to display social bookmarks. Possible values are `top` or `bottom`. The default value is `bottom`.
- **Social bookmarks types:** A comma-separated list of the possible bookmark types (twitter,facebook,plusone for example). Override with a `portal-ext.properties` file. The default value is `${server-property://com.liferay.portal/social.bookmark.types}`.

Knowledge Base Section Configuration

- **Show KBArticles sections title:** Sets whether to show the sections title in the application. This is enabled by default.
- **kb articles sections:** Sets the section portlet's title with a comma comma-separated list. The default value is `general`.
- **kb article display style:** Sets the default Knowledge Base article display style. Possible values are `title` or `abstract`. The default value is `title`.
- **Show KBArticles pagination:** Sets whether to show pagination in the application. This is enabled by default.
- **Enable KBArticle description:** Sets whether to display the description field in the edit view of the application. This is disabled by default.
- **Enable KBArticle ratings:** Sets whether to allow users to add ratings to the Knowledge Base Article. This is enabled by default.
- **Show KBArticle attachments:** Sets whether to show the attachments for the Knowledge Base article. This is displayed by default.
- **Show KBArticle asset entries:** Sets whether to show the asset entries for the Knowledge Base Article. This is displayed by default.
- **Show KBArticle asset links:** Sets whether to show the asset links for the Knowledge Base Article. This is displayed by default.
- **Enable KBArticle view count increment:** Sets whether to add a view counter to the Knowledge Base article. This is enabled by default.
- **Enable KBArticle subscriptions:** Sets whether to allow user to subscribe to the Knowledge Base article. This is enabled by default.
- **Enable KBArticle history:** Sets whether to allow users to view the history for the Knowledge Base article. This is enabled by default.
- **Enable KBArticle print:** Sets whether to give users the option to print the Knowledge Base article. This is enabled by default.
- **Enable social bookmarks:** Sets whether to display social bookmarks when viewing a Knowledge Base article. This is disabled by default.
- **Social bookmarks display style:** Sets the display style of the social bookmarks. Possible values are `menu`, `simple`, `vertical`, or `horizontal`. The default value is `menu`.
- **Social bookmarks display position:** Sets the position to display social bookmarks. Possible values are `top` or `bottom`. The default value is `bottom`.
- **Social bookmarks types:** A comma-separated list of the possible bookmark types (`twitter,facebook,plusone` for example). Override with a `portal-ext.properties` file. The default value is `${server-property://com.liferay.portal/social.bookmark.types}`.

- **Admin KBArticle sections:** Sets the Knowledge Base article sections available to the portlet.
- **Admin KBArticle sections default:** Sets the default selected section for Knowledge Base articles.

Although some of these settings can be modified through the configuration menu, others require more effort to change. These settings are specified next.

Overriding Settings

Some of the configuration settings have values and must be overridden through either a portal properties file or a configuration file. For example, the Knowledge Base Article configuration has the value shown below for the *Social bookmarks types* setting:

```
${server-property://com.liferay.portal/social.bookmark.types}
```

The `server-property` prefix specifies that this value is set through a server property. You can view the default value by going to *Configuration → Server Administration* and selecting the *Properties* tab and *Portal Properties* sub-tab. From here you can Search for the property you want(e.g., `social.bookmark.types`) to see the default value.

To override server properties, add the updated property values to the `portal-ext.properties` file in your app server's root directory.

The second kind of variable prefix is `resource`. For example, the *Email KBArticle suggestion in progress body* setting for the Knowledge Base Service configuration has the following value:

```
${resource:com/liferay/knowledge/base/dependencies/email_kb_suggestion_in_progress_body.tmpl}
```

This points to the resource that provides the template. To override these resources, you must change the configuration in the System Settings UI to point to your new file, or using a configuration file in your app server's `osgi/config` directory, override the property to point to the new file's absolute file path.

Note that email templates can also be modified through the Knowledge Base app's *Configuration* button in the Options menu.

Note: Once you've exported the configuration files, as explained in the System Settings User Guide documentation, the configuration files are located in a folder called `liferay-system-settings`. Within this folder you'll find the configuration(.config) files, named after the full class name for that file.

For example, the *Knowledge Base Service* configuration file is called `com.liferay.knowledge.base.configuration.KBGroupServ` and the *Knowledge Base Section* configuration file is called `com.liferay.knowledge.base.web.configuration.KBSectionPortletIns`

To override a configuration file, you must create a new configuration file with the same name, specify your property values, and place it in the `osgi/config` directory.

You can view the current configuration files and settings by exporting the configuration file as explained in the System Settings User Guide documentation.

Equivalent Configuration Properties for 6.2 Portlet Properties

The table below compares the 6.2 `portlet.properties` to the configuration properties and equivalent System Settings for the current version of the Knowledge Base. Although you can modify System Settings with a configuration file, it is recommended that you update them through the System Settings menus if possible.

The properties below are for the Knowledge Base Service System Settings configuration file:

Knowledge Base Service

File
Value
6.2 Portlet Property Configuration Property System Setting
admin.email.from.name emailFromName Email from name
6.2 Portlet Property Configuration Property System Setting
admin.email.from.address emailFromAddress Email from address
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.added.body emailKBArticleAddedBody Email KBArticle added body
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.added.enabled emailKBArticleAddedEnabled Email KBArticle added enabled
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.added.subject emailKBArticleAddedSubject Email KBArticle added subject
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.in.progress.body emailKBSuggestionInProgressBody Email KBArticle suggestion in progress body
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.in.progress.enabled emailKBSuggestionInProgressEnabled Email KBArticle suggestion in progress enabled
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.in.progress.subject emailKBSuggestionInProgressSubject Email KBArticle suggestion in progress subject
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.received.body emailKBSuggestionReceivedBody Email KBArticle suggestion received body
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.received.enabled emailKBSuggestionReceivedEnabled Email KBArticle suggestion received enabled
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.received.subject emailKBSuggestionReceivedSubject Email KBArticle suggestion received subject
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.resolved.body emailKBSuggestionResolvedBody Email KBArticle suggestion resolved body
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.resolved.enabled emailKBSuggestionResolvedEnabled Email KBArticle suggestion resolved enabled
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.suggestion.resolved.subject emailKBSuggestionResolvedSubject Email KBArticle suggestion resolved subject
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.updated.body emailKBSuggestionUpdatedBody Email KBArticle suggestion updated body
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.updated.enabled emailKBSuggestionUpdatedEnabled Email KBArticle suggestion updated enabled
6.2 Portlet Property Configuration Property System Setting
admin.email.kb.article.updated.resolved.subject emailKBSuggestionUpdatedSubject Email KBArticle suggestion updated subject

6.2 Portlet Property Configuration Property System Setting
admin.kb.article.increment.priority.enabled articleIncrementPriorityEnabled Article increment priority enabled
6.2 Portlet Property Configuration Property System Setting
markdown.importer.article.extensions markdownImporterArticleExtensions Markdown importer article extensions
6.2 Portlet Property Configuration Property System Setting
markdown.importer.article.intro markdownImporterArticleIntro Markdown importer article intro
6.2 Portlet Property Configuration Property System Setting
markdown.importer.image.file.extensions markdownImporterImageFileExtensions Markdown importer image file extensions
6.2 Portlet Property Configuration Property System Setting
markdown.importer.image.folder markdownImporterImageFolder Markdown importer image folder
6.2 Portlet Property Configuration Property System Setting
knowledge.base.source.url.enabled sourceURLEnabled Source URLEnabled
6.2 Portlet Property Configuration Property System Setting
knowledge.base.source.url.edit.message.key sourceURLEditMessageKey Source URLEdit message key
The properties below are for the Knowledge Base Section configuration file:
Knowledge Base Section
File
Value
6.2 Portlet Property Configuration Property System Setting
admin.kb.article.sections adminKBArticleSections Admin KBArticle sections
6.2 Portlet Property Configuration Property System Setting
admin.kb.article.sections.default adminKBArticleSectionsDefault Admin KBArticle sections default
The properties below have been removed or relocated:

- **knowledge.base.ratings.number.of.stars:** This property has been removed since Knowledge Base now has a Ratings Configuration Framework. You can change the ratings type in the Instance Settings configuration menu under the *Social* tab. You can also configure the ratings type for an individual site under the *Social* tab of the Site Settings configuration menu.

- **knowledge.base.social.bookmarks.display.style:** This property can now be found under the *Social bookmarks display style* setting of each Knowledge Base app's configuration menu in System Settings.

You can view the full list of available configuration files and properties by exporting the configuration files, as explained in the System Settings User Guide documentation.

55.12 Shield Adapter Settings

The following is a complete list of the settings you can configure with Liferay's Shield adapter.

`password=liferay` :The String value used for the password part of the authentication token.

`requiresAuthentication=true` :Set this boolean to `false` to connect to an Elasticsearch cluster not secured by Shield (in other words, don't use authentication or encryption).

`requiresSSL=true` :Set this boolean to `false` to connect to an Elasticsearch cluster that's secured by Shield but doesn't have SSL configured (in other words, use authentication but not encryption).

`sslKeystoreKeyPassword=` :When `requiresSSL=true`, set this String to the value of your Elasticsearch's `shield.ssl.keystore.key_password` property. See here for more information.

`sslKeystorePassword=liferay` :When `requiresSSL=true`, set this String to the value of your Elasticsearch's `shield.ssl.keystore.password` property. See here for more information.

`sslKeystorePath=/path/to/keystore.jks` :When `requiresSSL=true`, set this String to the value for your Elasticsearch's `shield.ssl.keystore.path` property. See here for more information.

`username=liferay` :The String value to use for the username part of the authentication token.

55.13 Solr Settings

Solr can be configured for use with 7.0. Liferay Marketplace includes a Solr adapter called Liferay Enterprise Search-Solr Search Engine. The adapter is configurable through System Settings or an OSGi configuration file named `com.liferay.portal.search.solr.configuration.SolrConfiguration.cfg` and deployed to `[Liferay_Home]/osgi/configs`.

The list below is all the configuration settings for Liferay's Solr adapter, in the order they appear in the System Settings application:

`authenticationMode=BASIC` A String with the value of *BASIC* or *CERT*. Use BASIC when connecting using the Basic Authentication plugin, otherwise select CERT to connect using 2-way SSL authentication.

`clientType=REPLICATED` A String with the value of *REPLICATED* or *CLOUD*. Use the default (REPLICATED) when connecting to a single-node Solr server. Specify CLOUD to connect to SolrCloud (see the next section, titled *High Availability with SolrCloud* for more information).

`logExceptionsOnly=true` A boolean value that, when set to true, only logs exceptions from Solr, without rethrowing them.

`readURL=http://localhost:8983/solr/liferay` A String array with the URLs to which Liferay will send search requests. This will be different from the `writeURL` if you use separate servers for indexing (write) and searching (read).

`writeURL=http://localhost:8983/solr/liferay` A String array with the URLs to which Liferay will send indexing requests. This will be different from the `readURL` if you use separate servers for indexing (write) and searching (read).

`zkHost=localhost:9983` A String with the ZooKeeper host and port. This is required when using the adapter in CLOUD mode.