Liferay®

Learning Paths - Level 01

Developer Setup

#LIFERAY**ENABLEMENT**

Session Breakdown

05 min	01	Liferay Workspace and Tools
05 min	02	Customizing and Extending Liferay
05 min	03	Building Client Extensions
05 min	04	What’s Next? + Q&A

Part 01

Liferay Workspace and Tools

Introduction to Liferay Workspace

- A Liferay Workspace is an environment specifically designed to manage your Liferay projects
- Typically represent a 1-to-1 where one project leverages one workspace
- Creates a standard structure and approach which is beneficial in team based environments
- Self encapsulating – can contain code and runtime
- Simplifies the process of building plugins and extensions through pre-configured tasks and automations
- Supports the full Lifecycle development: Create, Build, Test and Deploy
- Supports both Maven and Gradle (gradle is default)

Liferay Workspace

Benefits of Liferay Workspace

- **Increased Development Efficiency**: Streamlined project management, automation through Gradle, and pre-configured properties all contribute to faster development cycles and reduced development effort
- **Improved Collaboration**: Liferay Workspace facilitates collaboration among developers by providing a centralized platform for managing and sharing project resources.
- **Enhanced Consistency**: Standardized project structures and configurations within the workspace help maintain consistency across different Liferay modules and applications
- **Simplified Testing**: Workspace integration with Docker allows for efficient setup of various development, user acceptance testing, and production environments, facilitating thorough testing processes

Developer Tools

Liferay Workspace and Tools

Software

ESSENTIAL

- Java Development Kit (JDK) 11
- Gradle 7
- Git
- Liferay Cloud CLI*

* required for Liferay SaaS extension deployments

RECOMMENDED

- Blade CLI
- Liferay Cloud CLI
- Liferay Developer Studio
- IDEA Intellij
- Eclipse
- Database Server

ADDITIONAL

- Docker
- IDE Plugins

Liferay Workspace

Blade CLI

- Designed to streamline the development process for Liferay modules
- Simplifies the process of creating and managing Liferay modules
- Facilitates deploying modules to your Liferay server
- Supports interaction with the (Liferay) server to execute commands
- Generate sample projects based on specific module types, offering a quick starting point for development
- Sample commands:
  - Workspace creation: `blade init -v dxp-7.4-u102`
  - Liferay bundle download: `blade server init`
  - Liferay server management: `blade server run; blade server stop`
  - Module deployment: `blade deploy`

Liferay Workspace

Liferay Cloud CLI

- Designed specifically for managing your Liferay services deployed on the Liferay Cloud platform
- Allows developers and administrators to interact with their Liferay Cloud deployments remotely
- Supports common (near daily) activities for Liferay Cloud instances
  - Project and Service Management
  - Deployment and Updates
  - Log Management
  - Environment Management
  - Domain Management
- Sample commands:
  - Viewing projects and service statuses: `lcp list`
  - Deploying services from your repository: `lcp deploy`
  - Accessing a service containers shell: `lcp shell`

Part 02

Extending Liferay

Client Extensions

VS

Plugins

OSGI Plugins

- OSGi facilitates modular development and deployment within a container environment
- Written in Java and can interact with Liferay's core services
- Provide reusable functionalities that can be consumed by other bundles or the core Liferay application
- Benefit from Liferay's built-in security mechanisms and classloading infrastructure
- Intercept specific events within Liferay's lifecycle, allowing you to modify behavior or inject custom logic at various points
- High degree of integration with Liferay's core functionalities, enabling access to a wider range of services and APIs compared to Client Extensions
- WARNING: Can be highly complex and may require expert Liferay knowledge to be built safely and properly
- WARNING: Not available in Liferay SaaS

Plugins

Extending Liferay

Client Extensions

- Customize and extend the functionality of your Liferay portal without modifying the core Liferay code or using OSGi modules
- Operate outside of the Liferay application server and interact with Liferay through well-defined (Headless) APIs
- Offer simplified maintenance as all of the dependencies of Liferay are removed from the equation
- Versioned APIs can be used to handle evolution of portal without breaking extensions
- Eliminate need for expert knowledge of Liferay
- Cloud friendly; do not require modifications of underlying server
- Not limited by Liferay tech stack
- Supported by all deployment models (self hosted, PaaS and SaaS)

Client Extensions

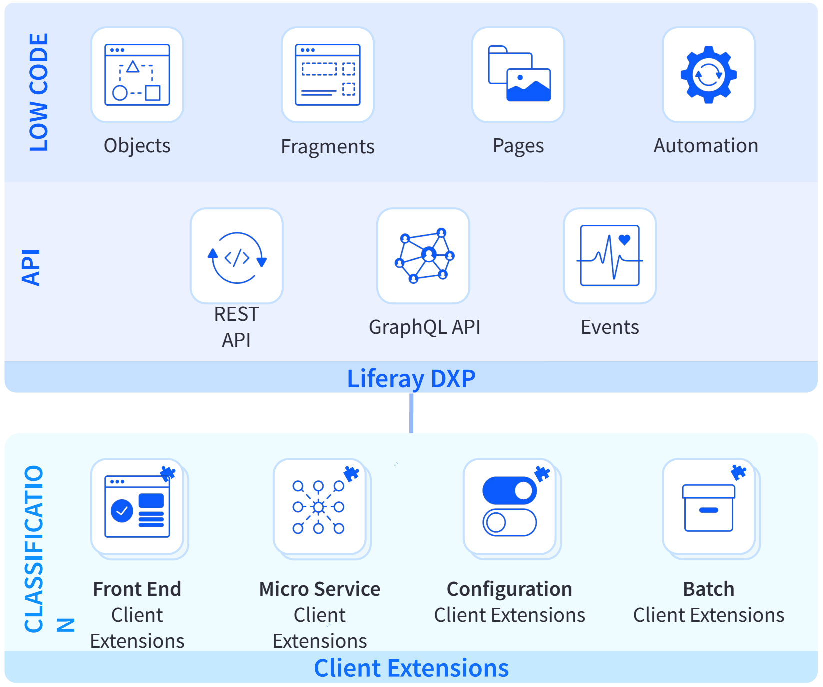
Part 03

Building Client Extensions

Extending Liferay

Client Extension Use Cases

- Client extensions are classified into 4 categories
  - Batch: provides data entities
  - Front-end: provides [generally static] resources
  - Microservice: provides API endpoints from Liferay that can run functions outside of Liferay
  - Configuration: provides configurations to change configure your Liferay instance
- Data/resource retrieval, API integrations, and Liferay instance configuration cover most Liferay use cases.
- Additional capability can be accessed through creating custom OSGi modules



Liferay Sample Workspace

- The Liferay Sample Client Extensions Workspace accelerates your client extension development
  - Naming conventions
  - Documentation
  - Example of every client extension
- Copy sample and rename to get started with your own client extension
- Test and deploy using Liferay tooling:
  - Blade CLI or IDE to deploy locally
  - Cloud CLI to deploy to SaaS

Liferay

Liferay workspace–accelerating development, streamlining success

Thank you