Liferay

Exercises for

Mastering Liferay Client Extensions

SKO Technical Clinic

Exercises for Mastering Liferay Client Extensions - SKO 2025 Edition

Table of Contents

- Exercise 1: Setting Up the SKO Workspace
- Exercise 2: Exporting the Contact Us Object Definition
- Exercise 3: Creating and Deploying a Batch Client Extension
- Exercise 4: Deploying Clarity's Ticketing Application

Exercise 1: Setting Up the SKO Workspace

Throughout the technical clinic, you'll use a local Liferay workspace for the hands-on exercises and practice what you learn. For that purpose, ensure you've completed the SKO Technical Clinic Prerequisites sent via email.

1. Open your terminal and run this command to verify Git is installed:

   ```
   git version
   ```

   Note: If you're on Windows, use Command Prompt, PowerShell, or BASH to execute terminal commands.

   This returns the version of your git installation. For example,

   ```
   git version 2.45.2
   ```

   If the Git command isn't found, please see official documentation for how to install Git for your OS (macOS | Windows | Linux/Unix).

2. Verify Java JDK 21 is installed:

   ```
   java -version
   ```

   The JDK version should display:

   ```
   openjdk version "21.0.5" 2024-10-15 LTS
   OpenJDK Runtime Environment Zulu21.38+21-CA (build 21.0.5+11-LTS)
   OpenJDK 64-Bit Server VM Zulu21.38+21-CA (build 21.0.5+11-LTS, mixed
   mode, sharing)
   ```

If Java isn't installed, you can find the appropriate OpenJDK distribution installer for your OS here. Alternatively, you can download the JDK as a ZIP (Windows) or TAR.GZ (Linux/Mac) package. To install, extract the file in a folder of your choice, then set the JAVA_HOME environment variable to that folder.

Note: If you support multiple Liferay projects and need to switch between different JDK versions, consider using a Version Manager:

- **Unix-based systems**:
  - SDKMAN!
  - jEnv
- **Windows**:
  - Jabba
  - JVMS

3. (Optional) Verify Blade is installed:

```
blade version
```

It should return the CLI's version:

```
blade version 6.0.0.202404102137
```

If Blade isn't installed, see Blade CLI installation instructions.

If the output indicates there's a newer version, run this command to update it:

```
blade update
```

Note: While we recommend using Blade to set up Liferay Workspace, you can also use Gradle to complete the process manually. See Creating a Liferay Workspace Manually for more information.

4. In your terminal, go to your desired folder and clone the training workspace to your computer:

```
git clone https://github.com/liferay/sko-2025
```

This saves a copy of the project in your current terminal directory.

Note: If you've cloned the repo previously, ensure your workspace is up to date by running `git pull origin main`.

5. Go to the workspace's root folder in your terminal:

```
cd sko-2025/
```

6. Initialize your Liferay bundle:

   ```
   blade server init
   ```

   This downloads and builds dependencies for running Liferay, including the Liferay server.

7. Use Blade to start your Liferay server:

   ```
   blade server run
   ```

   Alternatively,

   Unix-based:

   ```
   ./bundles/tomcat/bin/catalina.sh run
   ```

   Windows:

   ```
   .\bundles\tomcat\bin\catalina.bat run
   ```

   Tip: Wait until you see `org.apache.catalina.startup.Catalina.start Server startup in X milliseconds` to indicate startup completion.

8. When finished, access your Liferay DXP instance by going to `http://localhost:8080/` in your browser.

9. Sign in using these credentials:

   - Username: `admin@clarityvisionsolutions.com`
   - Password: `learn`

10. Take some time to explore the site and resources included in the training workspace.

Great! With your environment set up, you're ready to start contributing to Clarity's solutions!

Exercise 2: Exporting the Contact Us Object Definition

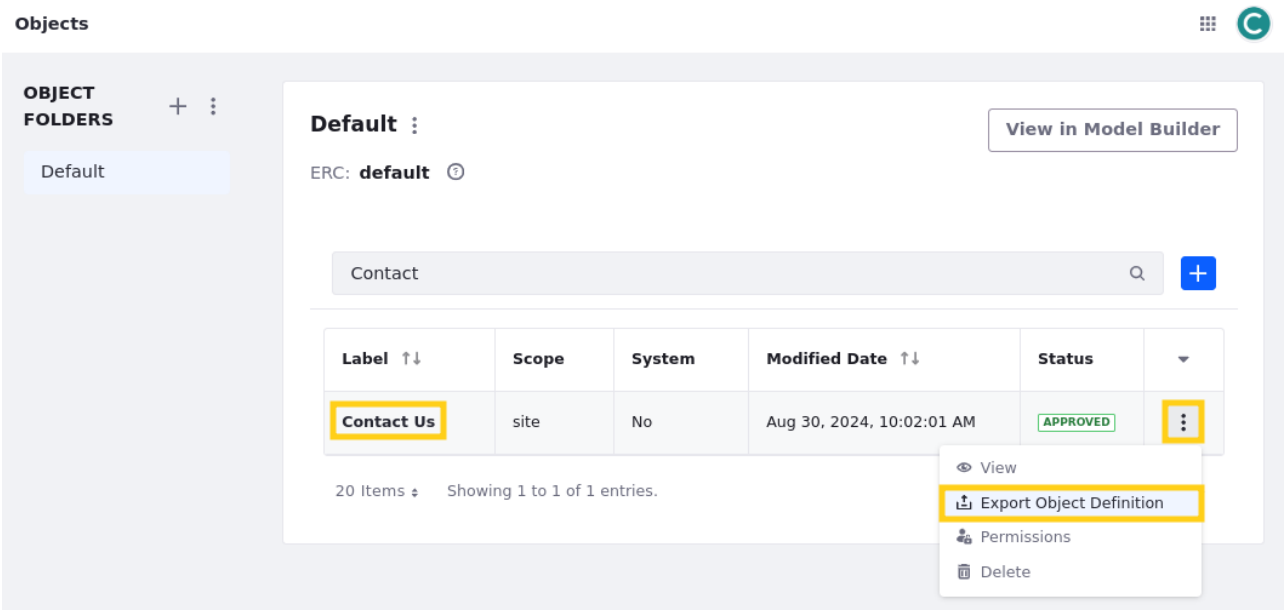
Here, you'll export Clarity's Contact Us object definition and explore its associated JSON file.

1. In your running Liferay instance, sign in as the Clarity Admin user.

   - Username: `admin@clarityvisionsolutions.com`
   - Password: `learn`

2. Open the *Global Menu* (), go to the *Control Panel* tab, and click *Objects*.

3. Click *Actions* ( ) for the Contact Us object and select *Export Object Definition*.



4. Open the downloaded `Object_Definition_ContactUs_[...].json` file in a text editor or IDE.

   Note: Many text editors and IDEs offer extensions to "prettify" JSON code, improving its readability for human comprehension.

5. Examine the file's JSON elements and nested values.

Great! You have successfully exported one of Clarity's object definitions and have learned a bit about its JSON structure. While this exercise leveraged the Contact Us object to demonstrate the necessary process to prepare for creating a batch client extension, Clarity's development team has created a Ticketing app on a lower environment that they need to migrate to other environments. Next, you'll learn how to package the JSON for Clarity's Ticketing app into a batch client extension.

Exercise 3: Creating Clarity's Ticketing Batch Client Extension

Here, you'll create a batch client extension containing the definition and related resources for Clarity's Ticketing app.

1. Open a file explorer and navigate to the `exercises/exercise-3/` folder in your course workspace.

2. Rename the `liferay-sample-batch` folder to `liferay-clarity-ticket-batch`.

   Note: The `liferay-sample-batch` client extension was downloaded from the Liferay Sample Workspace. As a best practice, use examples within this workspace as the baseline for your own client extension projects, as this serves as the primary source of truth for client extension implementation.

3. Navigate to the `liferay-clarity-ticket-batch/` folder.

4. Delete all files within the `batch` folder.

   This removes the sample client extension data to accommodate Clarity's Ticketing app content.

5. Open the `client-extension.yaml` file in a text editor or IDE.

   You'll define the batch client extension configuration in this file.

1. From the `exercise-3` folder, open the `ticket-batch-configuration.txt` file.

   This file contains the necessary configuration for the Ticketing app client extension.

2. Compare both files to spot the differences.

3. Replace the `client-extension.yaml` file's existing content with the code in the `ticket-batch-configuration.txt` file.

4. Save the file.

5. From the `exercise-3` folder, move these files into the `liferay-clarity-ticket-batch/batch/` folder:

   - `00-list-type-definition.batch-engine-data.json`
   - `01-object-definition.batch-engine-data.json`
   - `02-object-relationship.batch-engine-data.json`
   - `03-object-entry.batch-engine-data.json`

   With this, the client extension will create a picklist, the Ticket object definition, a relationship, and some Ticket entries upon deployment.

6. Move the `liferay-clarity-ticket-batch/` folder into the `client-extensions/` folder of your course workspace.

Great! Now that you've fully configured the batch client extension and moved it to the appropriate workspace location, you can deploy it into your Liferay environment.

Exercise 4: Deploying Clarity's Ticketing Application

Here, you'll deploy the previous exercise's batch client extension to migrate Clarity's Ticketing app.

1. Open a terminal and navigate to the `client-extensions/liferay-clarity-ticket-batch/` in your course workspace.

2. Run this command to build and deploy the client extension:

   ```
   blade gw clean deploy
   ```

3. Verify it deploys successfully.

   ```
   2025-01-13 14:33:19.157 INFO  [fileinstall-directory-watcher]
   [BundleStartStopLogger:68] STARTED liferayclarityticketbatch_7.4.13
   [1484]
   ```

Now that you've deployed the batch client extension, you can examine the migrated data model.

4. In your Liferay instance, sign in as the Clarity Admin user.
    - Username: `admin@clarityvisionsolutions.com`
    - Password: `learn`
5. Open the *Global Menu* (), go to the *Control Panel* tab, and click *Objects*.
6. Verify that the `Ticket` object definition is present.
7. In the Global Menu (), go to the *Control Panel* tab and click *Picklists*.
8. Verify that these picklists were created:
    - Priorities
    - Regions
    - Resolutions
    - Statuses
    - Types
9. In the Global Menu (), go to the *Control Panel* tab and click *Tickets*.
10. Verify that the sample Ticket entries were created.

Great! You've successfully created and deployed the batch clarity extension to migrate Clarity's Ticketing app.