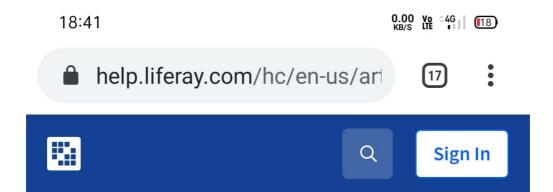


## **Spring MVC Portlet**

The Spring MVC portlet provides a way to add various different fields into the database and display them in a table. This project is a Spring MVC based portlet WAR that implements the same functionality as the apps/service-builder/basic-web sample project. It manages JSP pages for display, uses a Spring-annotated portlet class, and invokes the apps/service-builder/basic-api module to call services.

Note: If you're planning to package this sample using Maven, you must complete a few additional steps to avoid build errors. This sample relies on the service-builder/basic-api module.

Since the basic-api bundle is not available on Liferay's CDN repo or Maven Central, this sample can not reference it, resulting in build failures.



Now you can build this sample successfully.

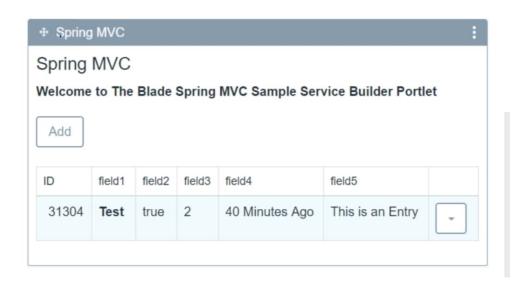


Figure 1: Click \*Add\* and fill out the sample fields to generate a custom entry in the portlet's table.

Unlike the service-builder/basic-web module, Spring MVC portlets must be delivered as portlet WAR projects. This project builds to a WAR file but leverages all of the Liferay Workspace tools and Gradle to build the WAR. You must build and deploy the service-builder/basic-api and service-builder/basic-service modules for this sample to work properly. For more information on using Spring MVC portlets in Liferay DXP, visit the Spring MVC tutorial



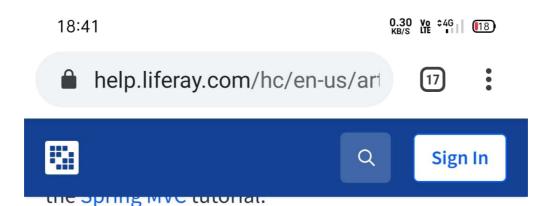
To satisfy this dependency, you must install the bundle dependency to your local ~/.m2 repo, along with the parent BND plugin and root Maven project. Here are the steps to accomplish this:

- 1. Run mvn clean install on
   maven/apps/service-builder/basicapi.
- 2. Run mvn clean install on
  maven/parent.bnd.bundle.plugin.
- 3. Run mvn clean install -N in the root liferay-blade-samples/maven folder.

Now you can build this sample successfully.



Figure 1: Click \*Add\* and fill out the sample fields to generate a custom entry in the portlet's table.



## What API(s) and/or code components does this sample highlight?

This sample demonstrates a Liferay DXP portlet built using the Spring Web MVC framework.

## How does this sample leverage the API(s) and/or code component?

You can easily modify this sample by customizing its SpringMVCPortletViewController Java class or any of its JSPs stored in the src/main/webapp/WEB-INF/jsp folder. For more information on customizing this sample, see the Javadoc listed in this sample's SpringMVCPortletViewController Java class.



can not reference it, resulting in build failures.

To satisfy this dependency, you must install the bundle dependency to your local ~/.m2 repo, along with the parent BND plugin and root Maven project. Here are the steps to accomplish this:

- 1. Run mvn clean install on
   maven/apps/service-builder/basicapi.
- 2. Run mvn clean install on
   maven/parent.bnd.bundle.plugin.
- 3. Run mvn clean install -N in the root liferay-blade-samples/maven folder.

Now you can build this sample successfully.



Figure 1: Click \*Add\* and fill out the sample fields