



APPLICATIONS: INTRODUCTION

Copyright ©2017 Liferay, Inc.

All Rights Reserved.

No material may be reproduced electronically or in print,
distributed, copied, sold, resold, or otherwise exploited
for any commercial purpose without express written
consent of Liferay, Inc.

TYPES OF APPLICATIONS IN LIFERAY DXP

- ❖ In Liferay, it is possible to create:
 - Liferay MVC Application
 - Traditional Model View Controller-style application that helps separate concerns of View, Render, Action, and Resource requests into different Java classes for more modularity.
 - Soy Portlet
 - You can look at the documentation on Soy Portlets for more information.
 - Spring MVC
 - Spring MVC is the web component of Spring's framework.
 - It provides a rich functionality for building robust Web Applications.
 - The Spring MVC Framework is designed so that every piece of logic and functionality is highly configurable.
 - In DXP, you can still use features like the Service Builder.
 - JavaServer Faces
 - Liferay Faces is an open-source umbrella project that provides support for the JavaServer™ Faces (JSF) standard in webapp and portlet projects.

CREATING AN APPLICATION USING BLADE CLI

- ❖ Blade CLI is a command line Java tool that can be used to help Bootstrap Liferay module development.
- ❖ It is installed using a Java package manager called jpm4j.
- ❖ Blade CLI makes it easy to create modules in Liferay.
- ❖ An MVC Application can be created with the following command:

```
blade create -t mvcportlet -p [package name] -c [class name] [project name]
```

- ❖ Here is a working example:

```
blade create -t mvcportlet -p com.liferay.docs.mvcportlet -c  
MyMvcPortlet my-mvc-portlet-project
```

FOLDER STRUCTURE

- ❖ From here, you can use the Liferay MVC Framework for creating your application.

```
- my-mvc-portlet-project
  - src
    - main
      - java
        - com.liferay/docs/mvcportlet
          - MyMvcPortlet.java
    - resources
      - content
        - Language.properties
      - META-INF
        - resources
          - css
          - js
          - init.jsp
          - view.jsp
  - bnd.bnd
  - build.gradle
```

START BUILDING

- ❖ Having the following code in the `MyMVCPortlet.java` is the starting point for building your application:

```
@Component(  
    immediate = true,  
    property = {  
        "com.liferay.portlet.css-class-wrapper=portlet-my-mvc-portlet",  
        "com.liferay.portlet.display-category=category.sample",  
        "com.liferay.portlet.footer-portlet-javascript=/js/main.js",  
        "com.liferay.portlet.header-portlet-css=/css/main.css",  
        "com.liferay.portlet.preferences-owned-by-group=true",  
        "com.liferay.portlet.private-request-attributes=false",  
        ...  
    },  
    service = Portlet.class  
)  
public class MyMvcPortlet extends MVCPortlet {  
    ...  
}
```

Notes: