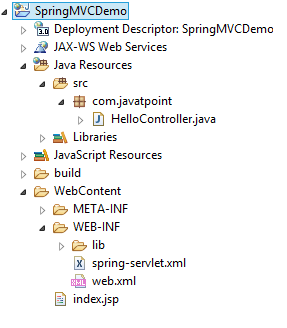
**LabSheet 12**

creating Spring web application to implement SpringMVC framework using eclipse IDE.

Problem Statement : to write a simple web-based Hello World application using Spring MVC framework.

# Creating a Spring Boot Project

Following are the steps to create a simple Spring Boot Project.



## **Step 1:**  Create a *Dynamic Web Project* with a name *SpringMVC* and create a package *com.pu*  under the *src* folder in the created project.

## **Step 2:** Drag and drop below mentioned Spring and other libraries into the folder *WebContent/WEB-INF/lib*.

Download Link : Download all jar files for spring https://static.javatpoint.com/src/sp/springjars.zip

Step 3: Create a Java class *HelloController* under the *com.pu* package.

1. **package** com.pu;
2. **import** org.springframework.stereotype.Controller;
3. **import** org.springframework.web.bind.annotation.RequestMapping;
4. @Controller
5. **public** **class** HelloController {
6. @RequestMapping("/")
7. **public** String display()
8. {
9. **return** "index";
10. }
11. }

Step4: Create Spring configuration files *web.xml* and *spring-servlet.xml* under the *WebContent/WEB-INF* folder.

**web.xml**

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<web-app** xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd" id="WebApp\_ID" version="3.0"**>**
3. **<display-name>**SpringMVC**</display-name>**
4. **<servlet>**
5. **<servlet-name>**spring**</servlet-name>**
6. **<servlet-class>**org.springframework.web.servlet.DispatcherServlet**</servlet-class>**
7. **<load-on-startup>**1**</load-on-startup>**
8. **</servlet>**
9. **<servlet-mapping>**
10. **<servlet-name>**spring**</servlet-name>**
11. **<url-pattern>**/**</url-pattern>**
12. **</servlet-mapping>**
13. **</web-app>**

**spring-servlet.xml**

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<beans** xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xmlns:context="http://www.springframework.org/schema/context"
5. xmlns:mvc="http://www.springframework.org/schema/mvc"
6. xsi:schemaLocation="
7. http://www.springframework.org/schema/beans
8. http://www.springframework.org/schema/beans/spring-beans.xsd
9. http://www.springframework.org/schema/context
10. http://www.springframework.org/schema/context/spring-context.xsd
11. http://www.springframework.org/schema/mvc
12. http://www.springframework.org/schema/mvc/spring-mvc.xsd"**>**
14. <!-- Provide support for component scanning -->
15. **<context:component-scan** base-package="com.javatpoint" **/>**
17. <!--Provide support for conversion, formatting and validation -->
18. **<mvc:annotation-driven/>**
20. **</beans>**

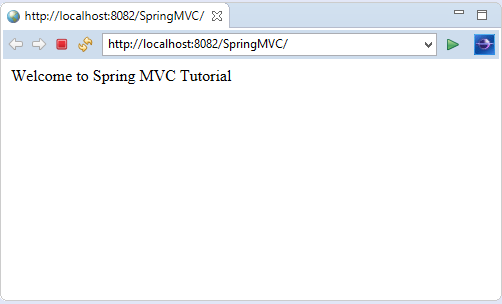
Step 5: Create file name *jsp* under the *WebContent/WEB-INF* folder. Create a view file *index.jsp*

**index.jsp**

1. **<html>**
2. **<body>**
3. **<p>**Welcome to Spring MVC Tutorial**</p>**
4. **</body>**
5. **</html>**

Step 6: The final step is to create the content of all the source and configuration files and export the application as explained below.

Once you are done creating the source and configuration files, export your application. Right-click on your application and use **Export > WAR File** option and save your **SpringMVC.war** file in Tomcat's *webapps* folder.



https://www.javatpoint.com/spring-mvc-tutorial