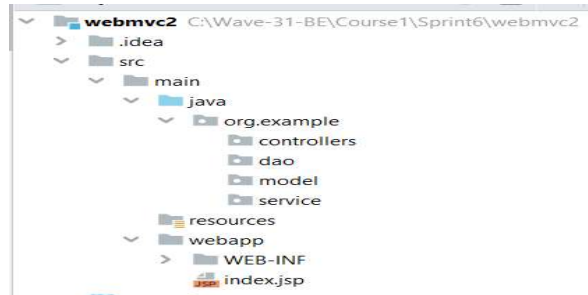


Steps to create spring mvc webapplication

1 Make maven webapp,add tomcat, run application as webapp

make sure index.jsp is visible in browser thru BASE URL of application

create required packages



2 add required dependencies in pom

spring-webmvc

jstl

javax-servlet

```
19 <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
20 <dependency>
21   <groupId>org.springframework</groupId>
22   <artifactId>spring-webmvc</artifactId>
23   <version>5.3.18</version>
24 </dependency>
25
26 <!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
27 <dependency>
28   <groupId>javax.servlet</groupId>
29   <artifactId>jstl</artifactId>
30   <version>1.2</version>
31 </dependency>
32
33 <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
34 <dependency>
35   <groupId>javax.servlet</groupId>
36   <artifactId>javax.servlet-api</artifactId>
37   <version>4.0.1</version>
38   <scope>provided</scope>
39 </dependency>
```

- 3 configure web.xml under 'WEB-INF' file
define front controller/dispatcher servlet configuration



The screenshot shows the 'web.xml' file in an IDE. The XML content is as follows:

```
1 <!DOCTYPE web-app PUBLIC
2   "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
3   "http://java.sun.com/dtd/web-app_2_3.dtd" >
4
5 <web-app>
6   <display-name>Archetype Created Web Application</display-name>
7
8   <!-- define dispatcher servlet info -->
9
10  <servlet>
11    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
12    <servlet-name>dispatcher</servlet-name>
13  </servlet>
14
15  <servlet-mapping>
16    <servlet-name>dispatcher</servlet-name>
17    <url-pattern>/</url-pattern>
18  </servlet-mapping>
19  <!-- any request after / in BASE URL, request has to be processed as per dispatcher servlet config -->
20
21 </web-app>
```

A blue callout box with a double-headed arrow points to the `<servlet-name>dispatcher</servlet-name>` tag in line 12 and the `<servlet-name>dispatcher</servlet-name>` tag in line 16. The text in the box reads: "frontcontroller xml filename must be 'dispatcher-servlet.xml'".

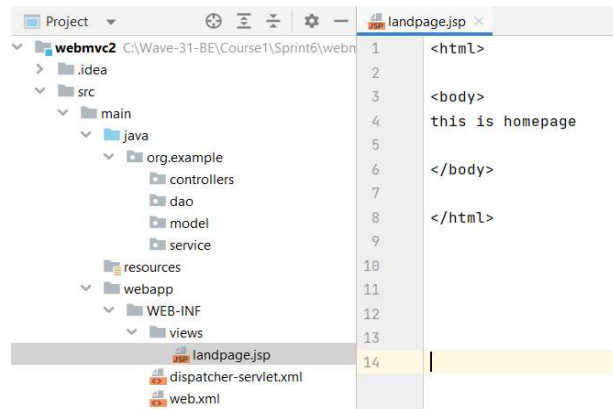
- 4 Create dispatcher-servlet.xml under 'WEB-INF' file
define required info



The screenshot shows the 'dispatcher-servlet.xml' file in an IDE. The XML content is as follows:

```
1 <beans xmlns = "http://www.springframework.org/schema/beans"
2   xmlns:context = "http://www.springframework.org/schema/context"
3   xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation = "http://www.springframework.org/schema/beans
5     http://www.springframework.org/schema/beans/spring-beans.xsd
6     http://www.springframework.org/schema/context
7     http://www.springframework.org/schema/context/spring-context.xsd">
8
9   <!-- location of controllers -->
10  <context:component-scan base-package="org.example.controllers"/>
11
12
13  <bean id="jspViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
14    <!-- which class to be used to resolve views from JSPs -->
15    <!-- location of views -->
16    <property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>
17    <property name="prefix" value="/WEB-INF/views"/>
18    <property name="suffix" value=".jsp"/>
19  </bean>
20
21 </beans>
```

5 Keep view files in proper folder



6 Create controller class

define request handler method to get landpage as first view
for default request

<http://localhost:8080/webmvc2/>

BASE_URL/

req hand

landpage.jsp

request

/

request handler method

HomeController.displayHomePage()

```
1 package org.example.controllers;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.RequestMapping;
5
6 @Controller // annotation makes below class as controller class
7 public class HomeController {
8
9     @RequestMapping("/")
10    public String displayHomePage(){
11        return "landpage";
12        // view is resolved for landpage.jsp, goes to browser as response
13    }
14
15 }
```