

## Lesson 03 Demo 01

### Create Spring MVC Web Project Structure

**Objective:** To create a Spring Web MVC application using the Eclipse IDE and configure the necessary components such as controllers, views, and the dispatcher servlet

**Tool required:** Eclipse IDE

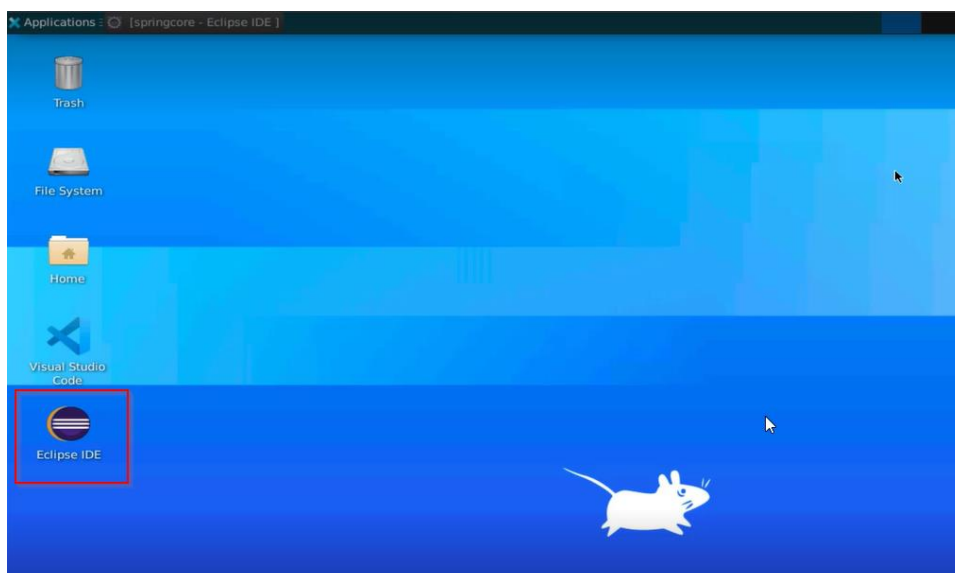
**Prerequisites:** None

#### Steps to be followed:

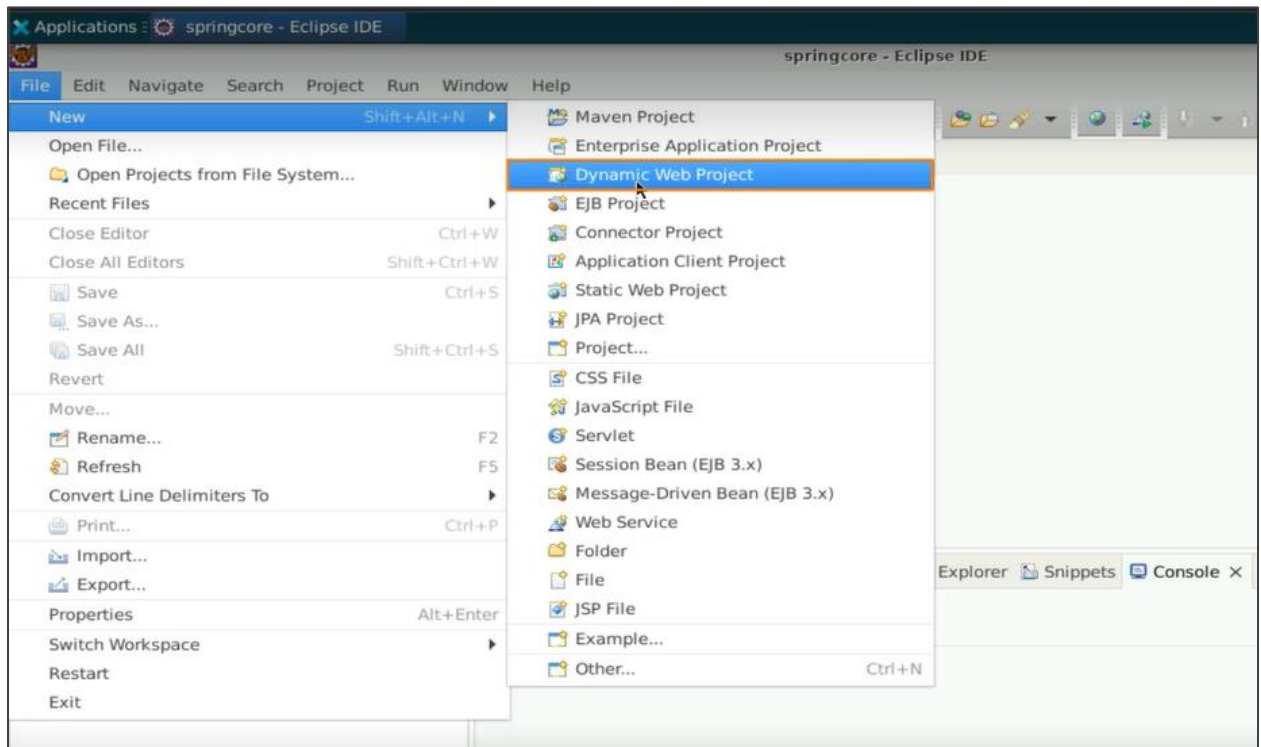
1. Creating a Dynamic Web Project
2. Converting the project to a Maven project and configuring dependencies
3. Creating a controller class
4. Creating views (JSP page) for the application
5. Configuring the DispatcherServlet

#### Step 1: Creating a Dynamic Web Project

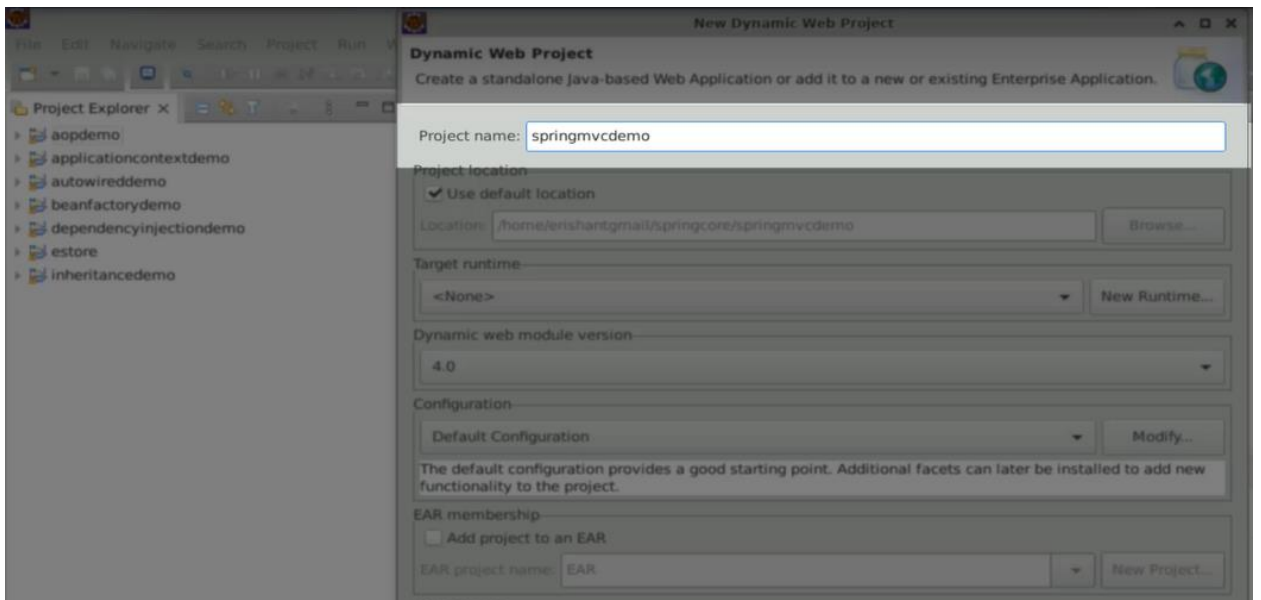
##### 1.1 Open Eclipse IDE



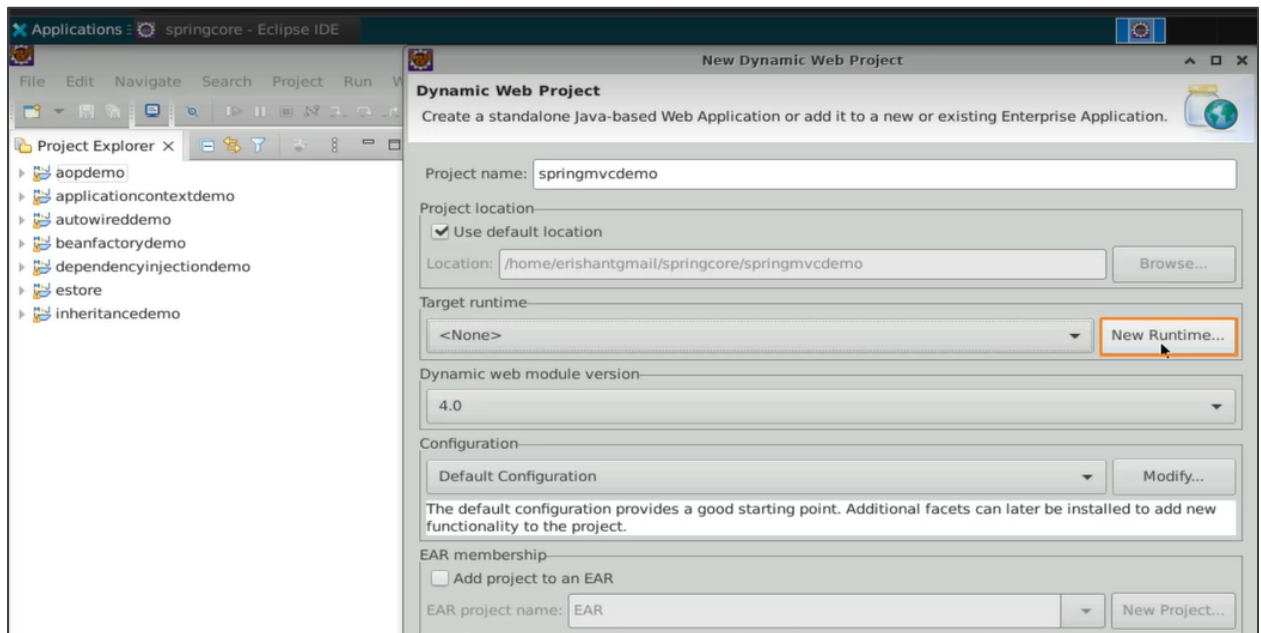
1.2 Click **File > New > Dynamic Web Project** to create a new web project



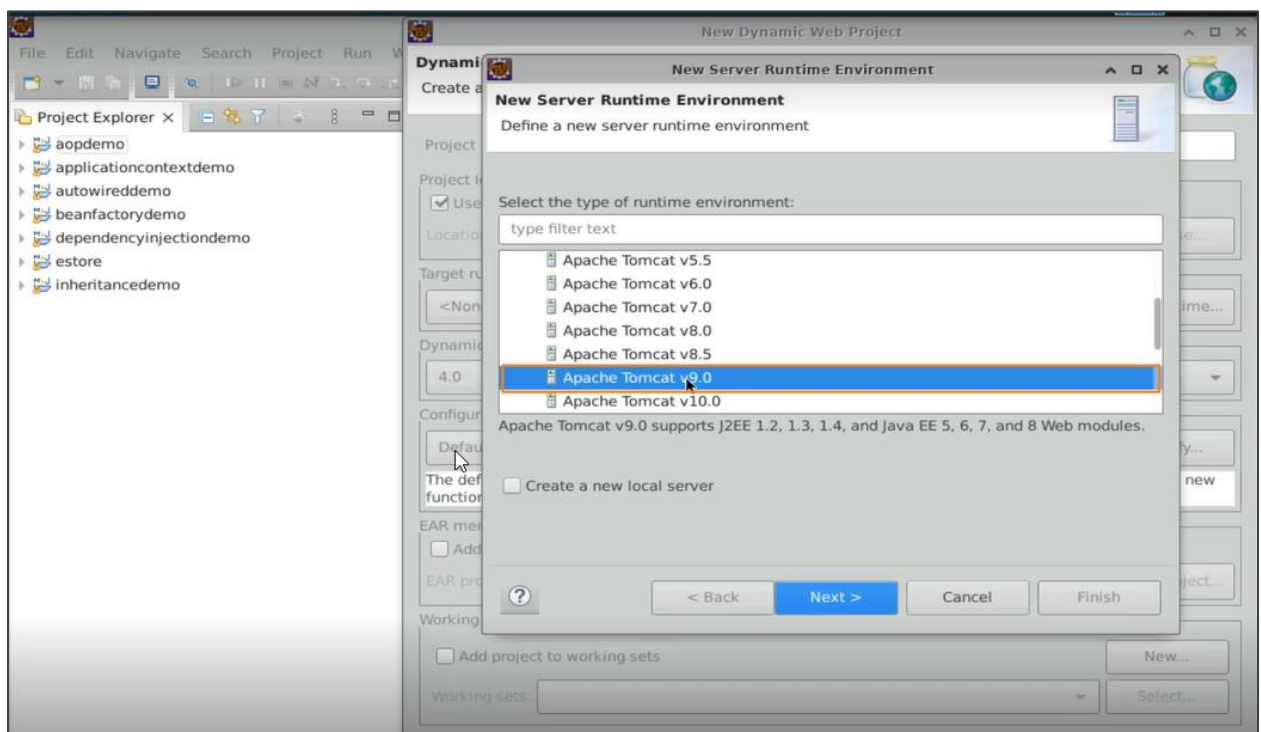
1.3 Name the project as **springmvcdemo**



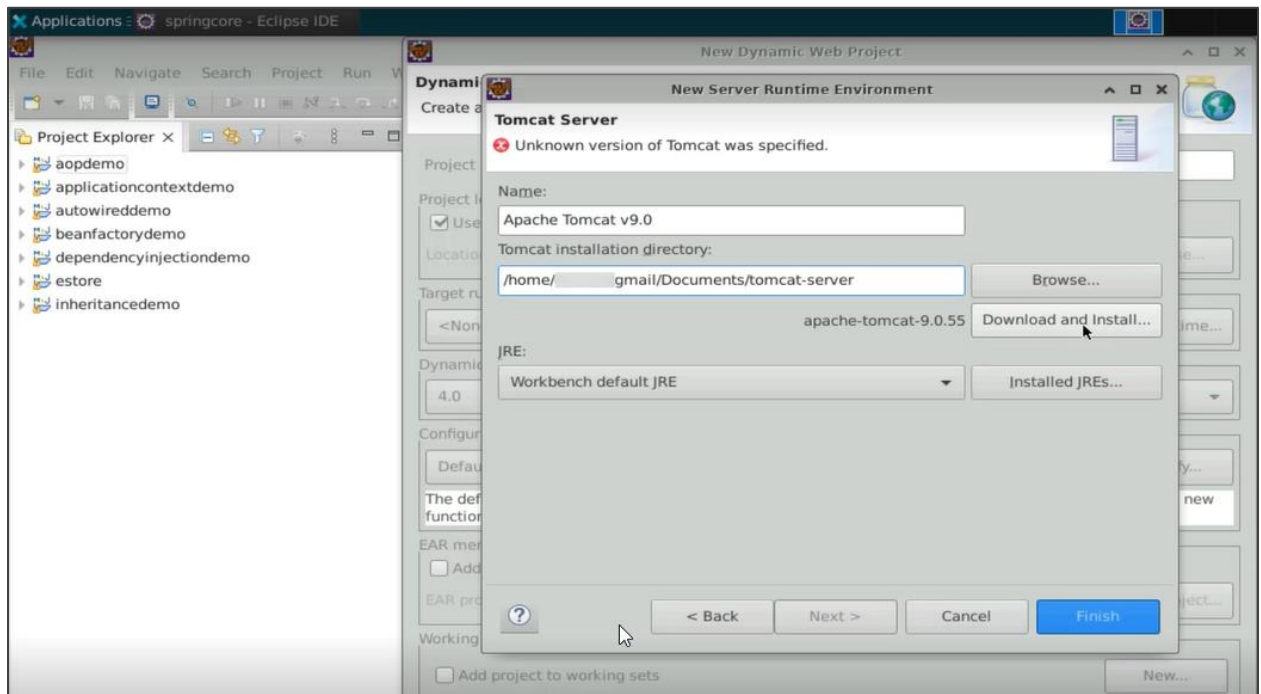
1.4 Next, click on **New Runtime** to add a new server for the application



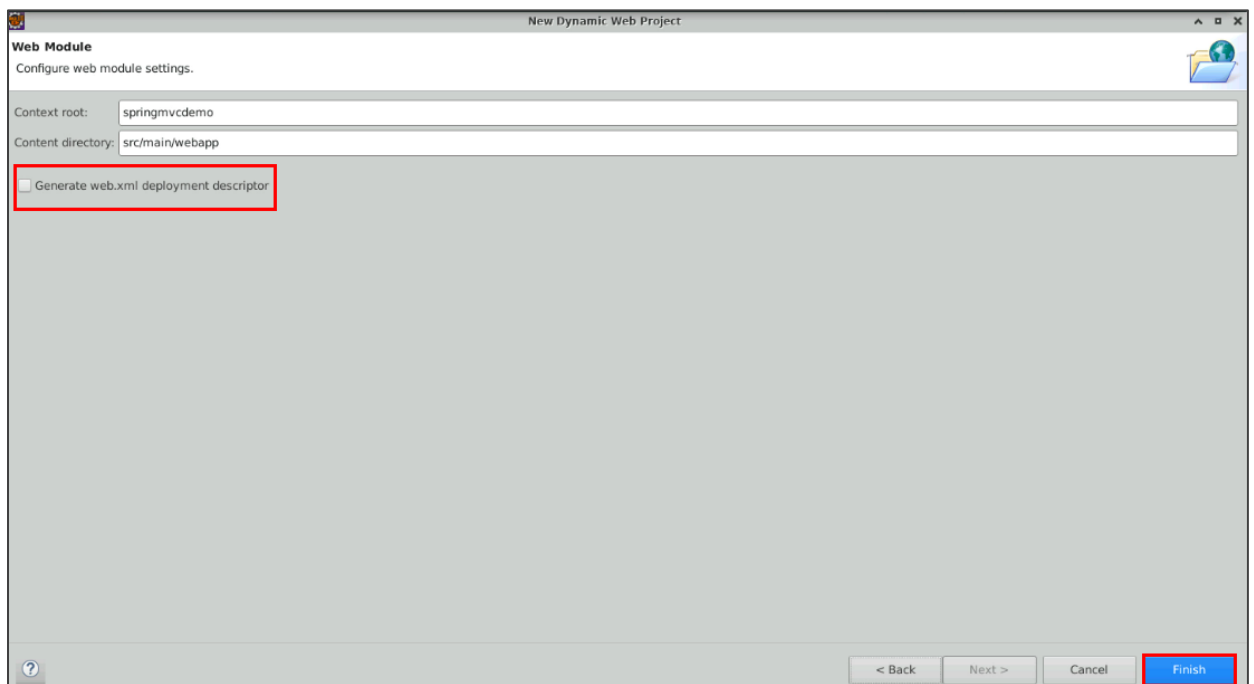
1.5 Choose the **Apache Tomcat v9.0** and click **Next**



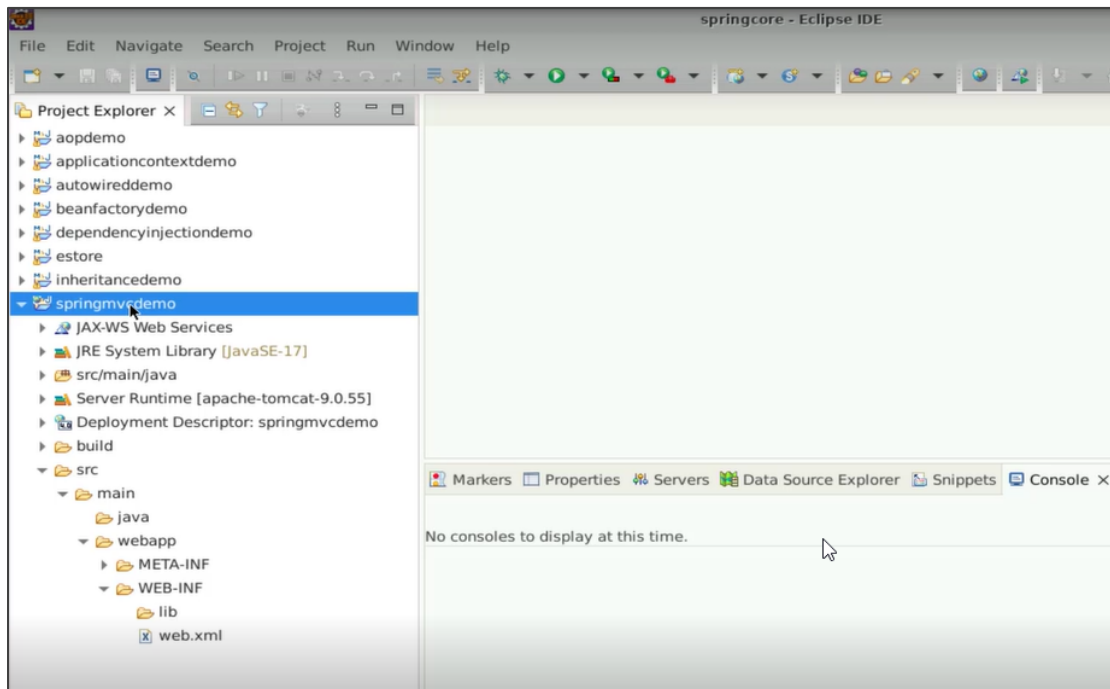
1.6 Provide the installation directory, select **Download and Install**, and click **Finish**



1.7 Select the option **Generate web.xml deployment description** and click **Finish**

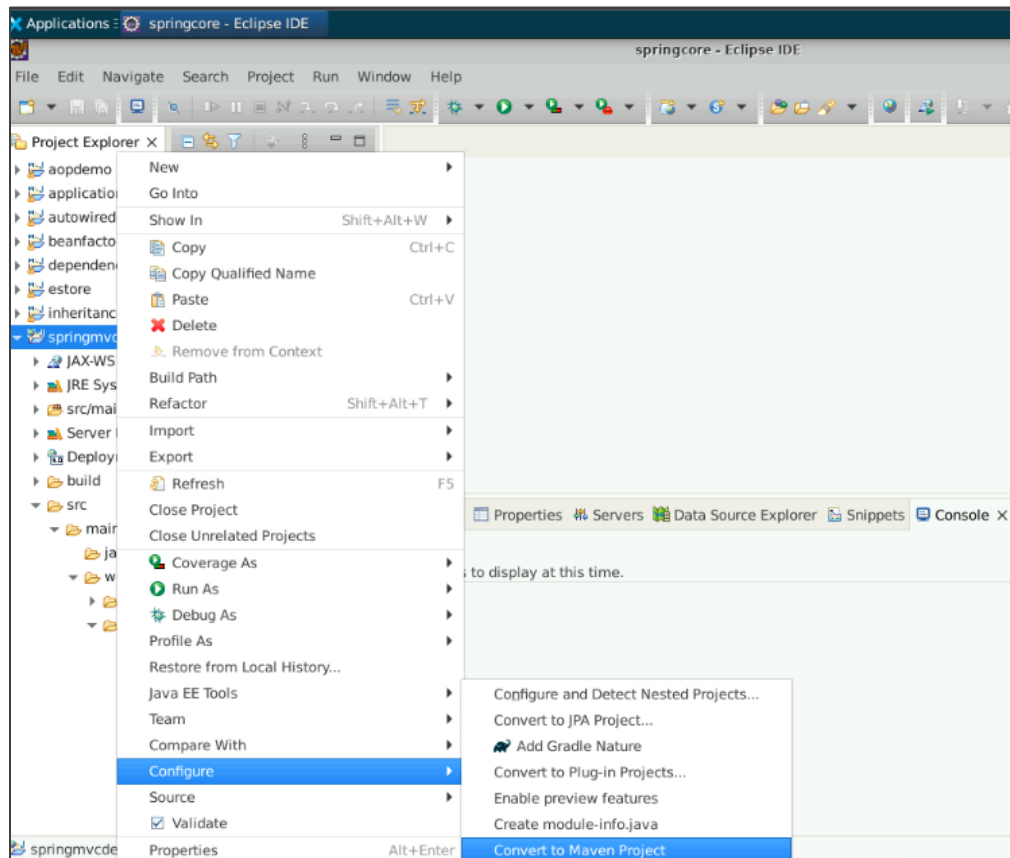


The project structure will appear as shown below:

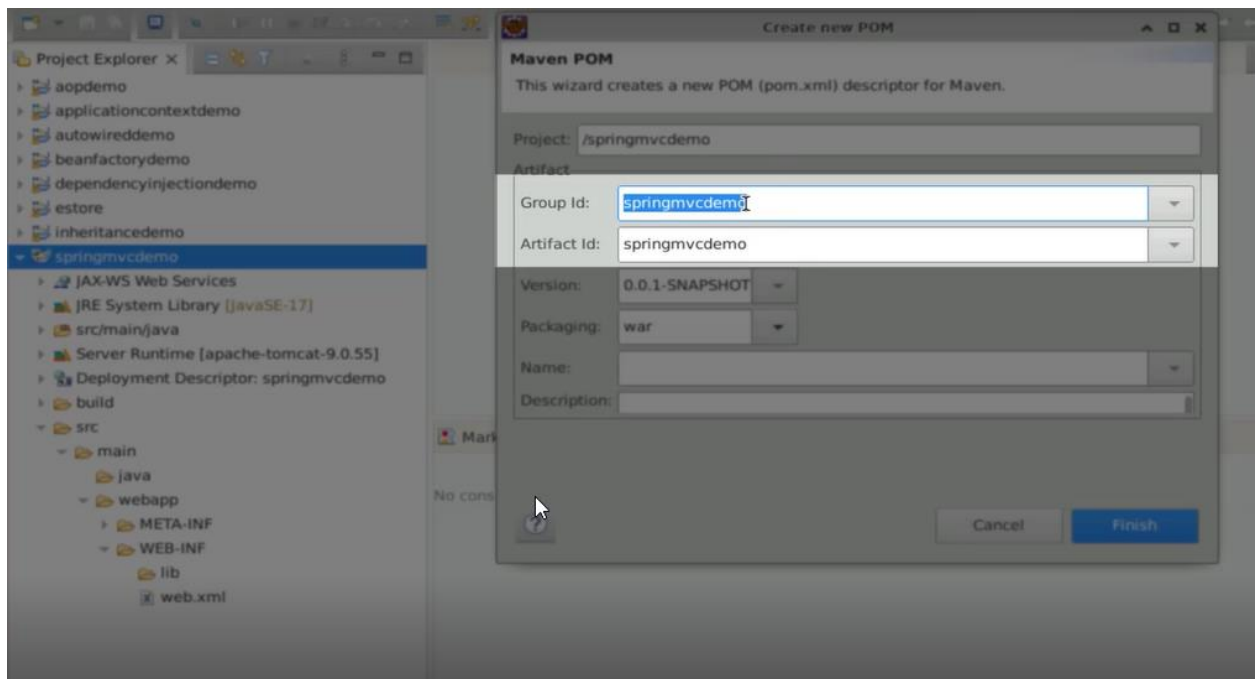


## Step 2: Converting the project to a Maven project and configuring dependencies

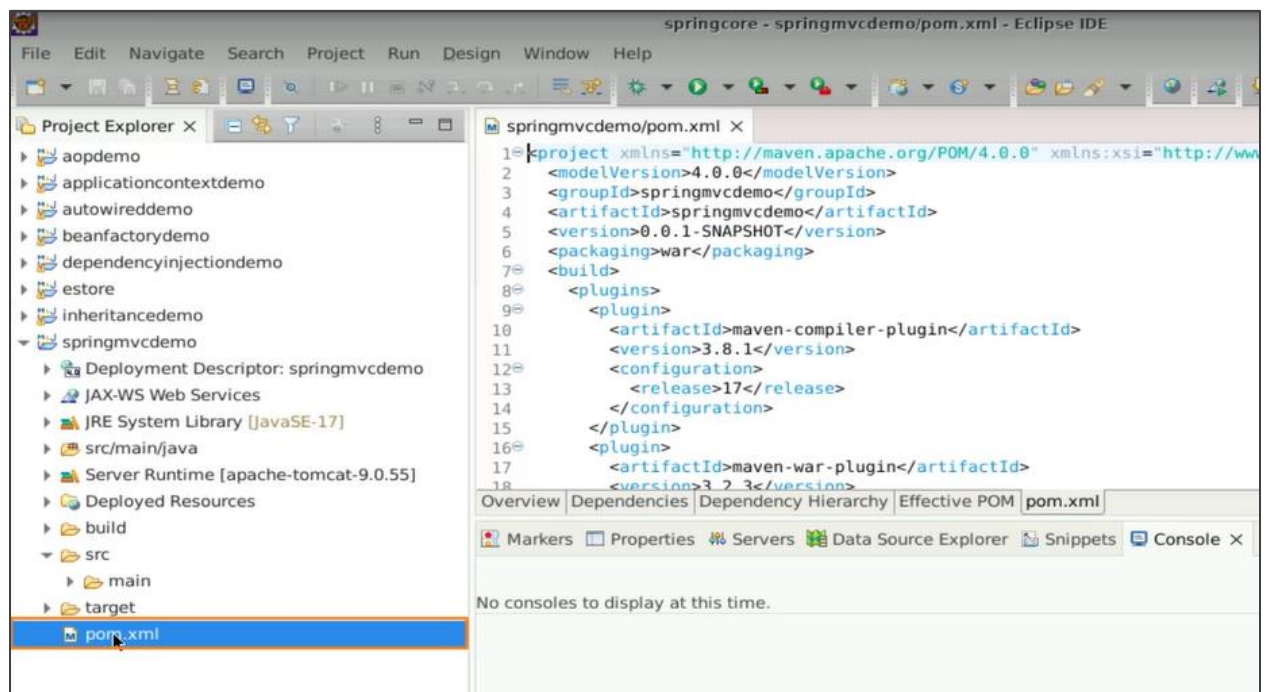
### 2.1 Right-click on the project and select **Configure > Convert to Maven Project**



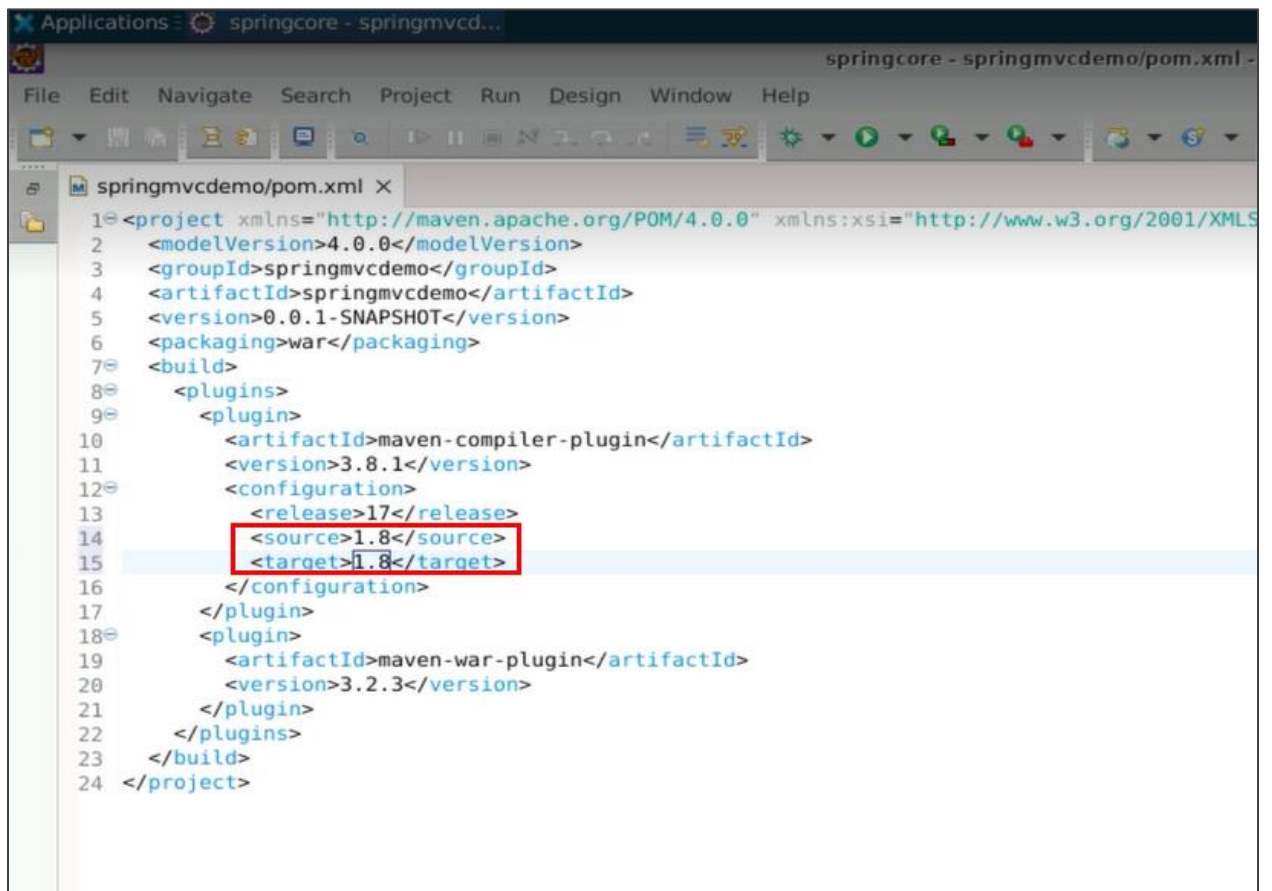
## 2.2 Choose the same Group and Artifact Id and click **Finish**



## 2.3 Open the **pom.xml** file in the project directory



## 2.4 Add <source> and <target> tags with the value set as 1.8



```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>springmvcdemo</groupId>
4   <artifactId>springmvcdemo</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <packaging>war</packaging>
7   <build>
8     <plugins>
9       <plugin>
10        <artifactId>maven-compiler-plugin</artifactId>
11        <version>3.8.1</version>
12        <configuration>
13          <release>17</release>
14          <source>1.8</source>
15          <target>1.8</target>
16        </configuration>
17      </plugin>
18      <plugin>
19        <artifactId>maven-war-plugin</artifactId>
20        <version>3.2.3</version>
21      </plugin>
22    </plugins>
23  </build>
24 </project>
```



## 2.5 Add the necessary dependencies for Spring Web and Spring Web MVC under the <dependencies> tag

```
*springmvcdemo/pom.xml X
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>springmvcdemo</groupId>
4   <artifactId>springmvcdemo</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <packaging>war</packaging>
7
8   <dependencies>
9     <dependency>
10       <groupId>org.springframework</groupId>
11       <artifactId>spring-web</artifactId>
12       <version>5.3.15</version>
13     </dependency>
14   </dependencies>
15
16   <build>
17     <plugins>
18       <plugin>
19         <artifactId>maven-compiler-plugin</artifactId>
20         <version>3.8.1</version>
21         <configuration>
22           <release>17</release>
23           <source>1.8</source>
24           <target>1.8</target>
25         </configuration>
26       </plugin>
27       <plugin>
28         <artifactId>maven-war-plugin</artifactId>
29         <version>3.2.3</version>
30       </plugin>
31     </plugins>
32   </build>
33 </project>
```

```
*springmvcdemo/pom.xml X
3   <groupId>springmvcdemo</groupId>
4   <artifactId>springmvcdemo</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <packaging>war</packaging>
7
8   <dependencies>
9     <dependency>
10       <groupId>org.springframework</groupId>
11       <artifactId>spring-web</artifactId>
12       <version>5.3.15</version>
13     </dependency>
14     <dependency>
15       <groupId>org.springframework</groupId>
16       <artifactId>spring-webmvc</artifactId>
17       <version>5.3.15</version>
18     </dependency>
19   </dependencies>
20
```

2.6 Save the **pom.xml** file and update the Maven project by right-clicking on the project and selecting **Maven > Update Project**

The following dependencies will be added to the project:

```

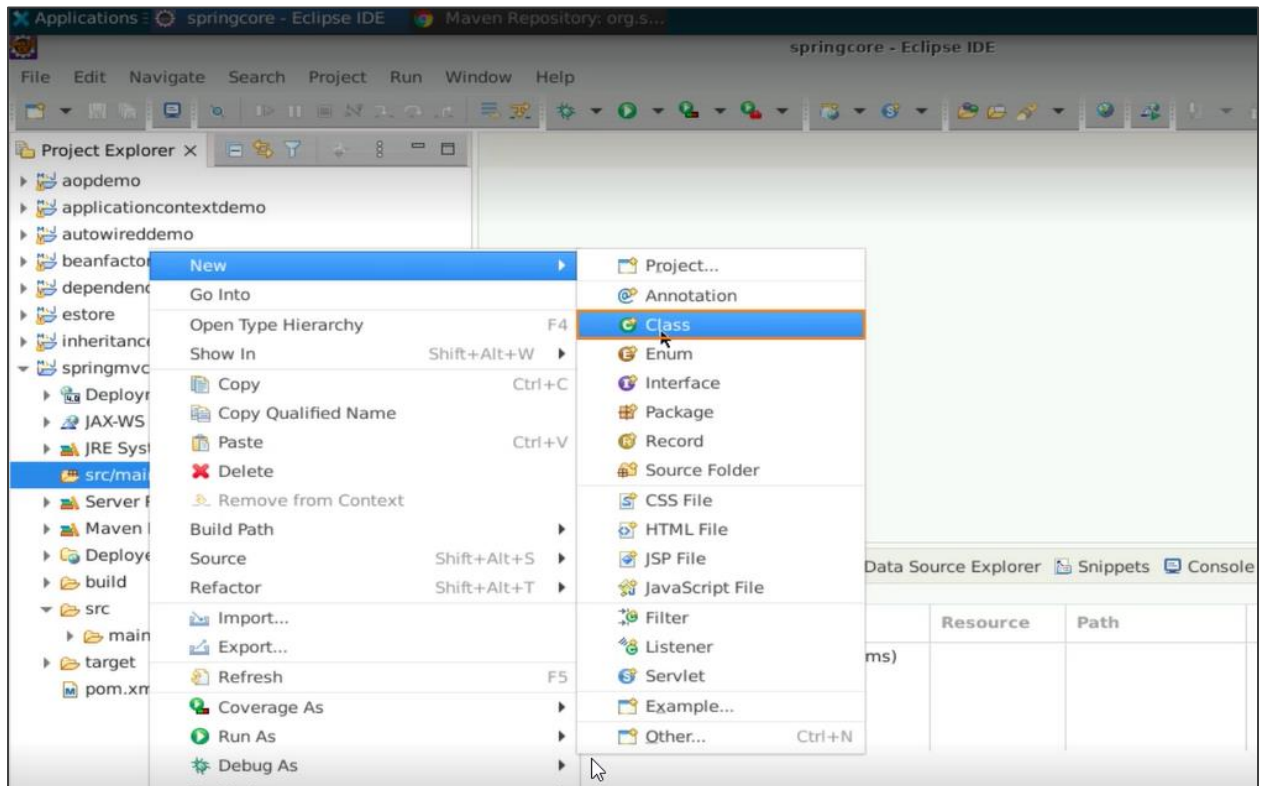
12     <version>5.3.15</version>
13   </dependency>
14
15   <dependency>
16     <groupId>org.springframework</groupId>
17     <artifactId>spring-webmvc</artifactId>
18     <version>5.3.15</version>
19   </dependency>

```

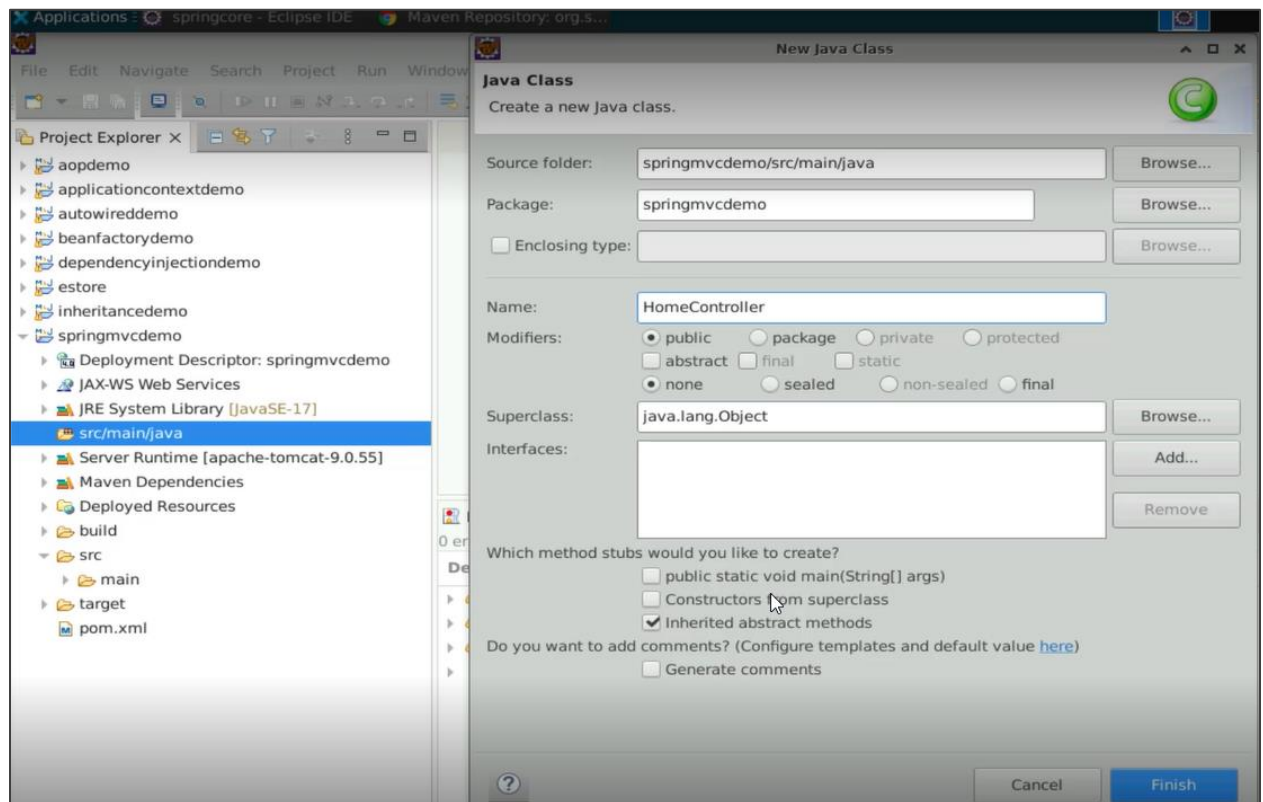
Description	Resource	Path
⚠ JRE Compiler Compliance Problem (7 items)		
⚠ Java Build Path Problems (7 items)		
⚠ Java Problems (8 items)		
⚠ Java Task (1 item)		

## Step 3: Creating a controller class

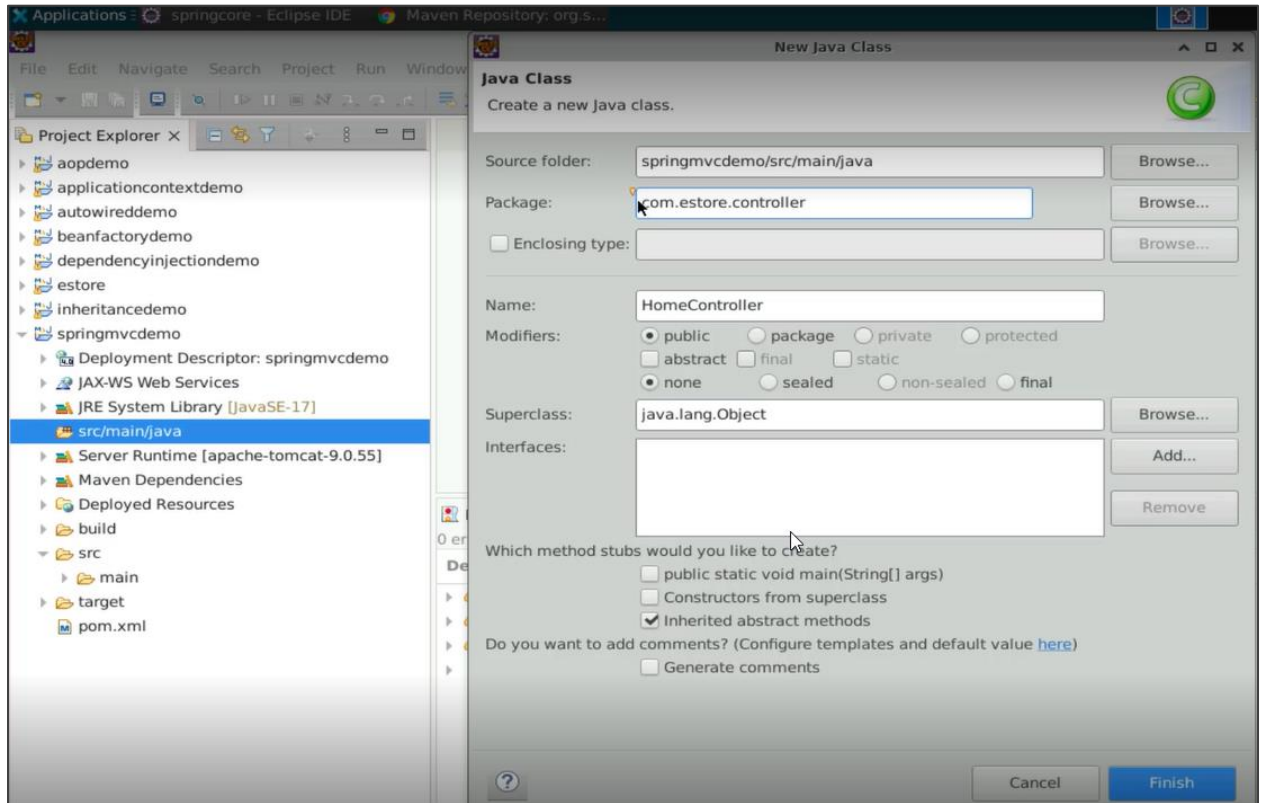
### 3.1 Right-click on the source folder in the project and select **New > Class**



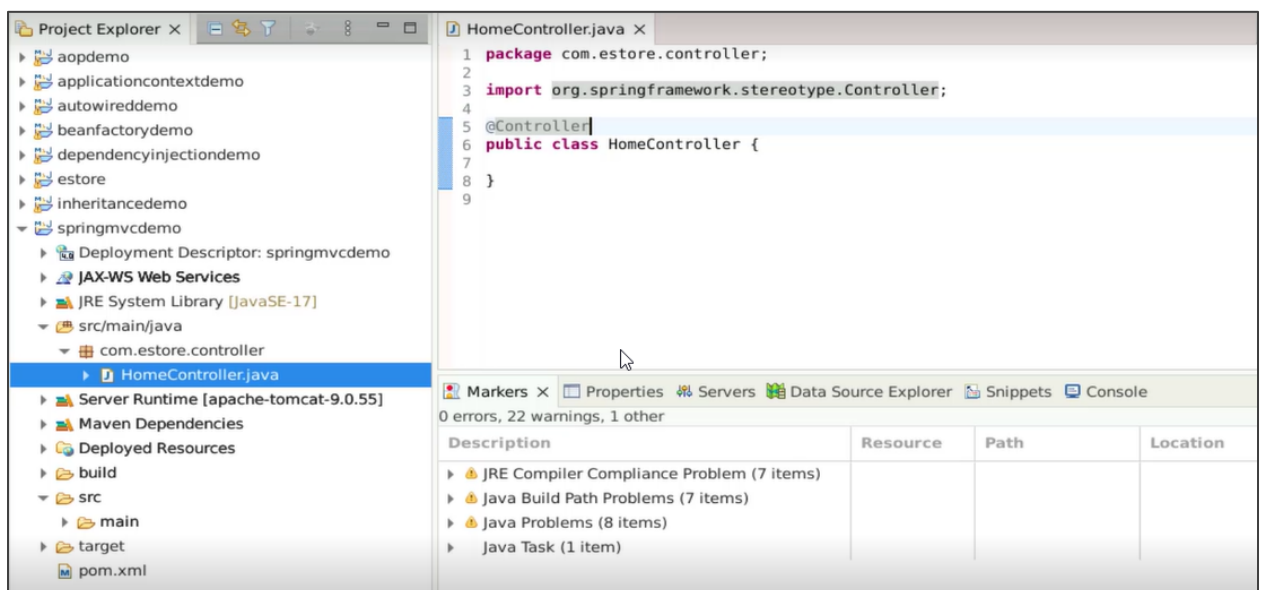
### 3.2 Enter a name for the class, such as **HomeController**



3.3 Specify the package name, for example, **com.estore.controller**. Click **Finish** to create the class

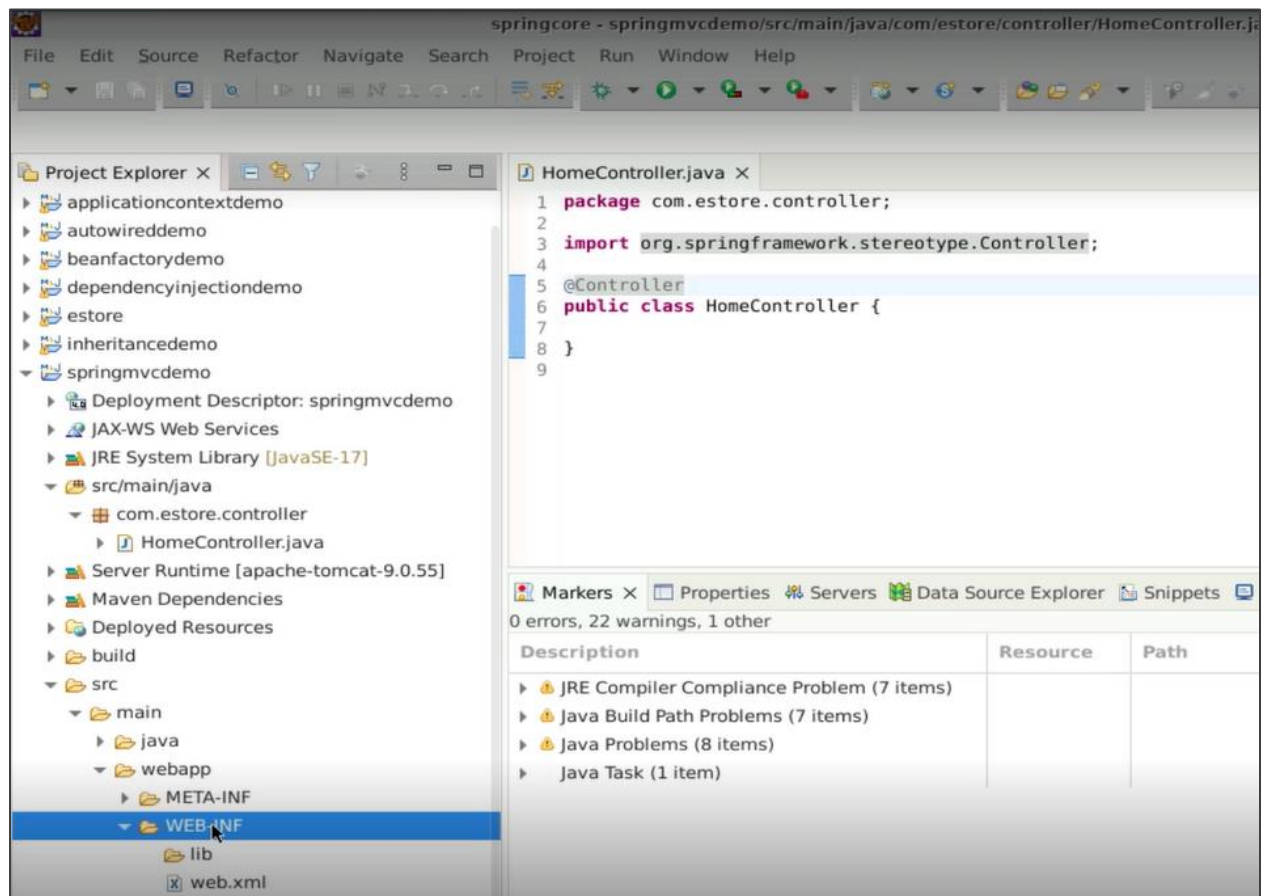


3.4 Inside the HomeController class, add the **@Controller** annotation

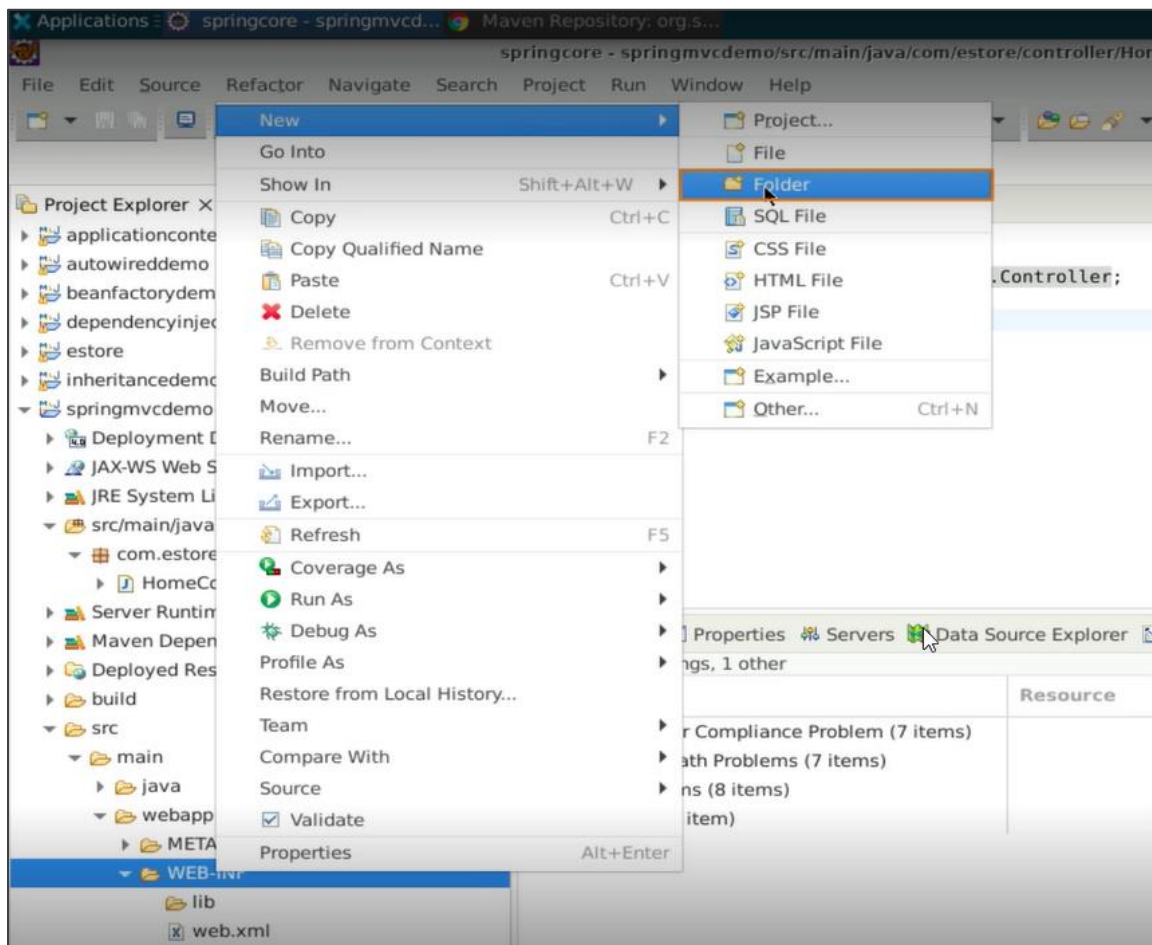


## Step 4: Creating views (JSP page) for the application

4.1 In the project directory, navigate to the **webapp/WEB-INF** folder

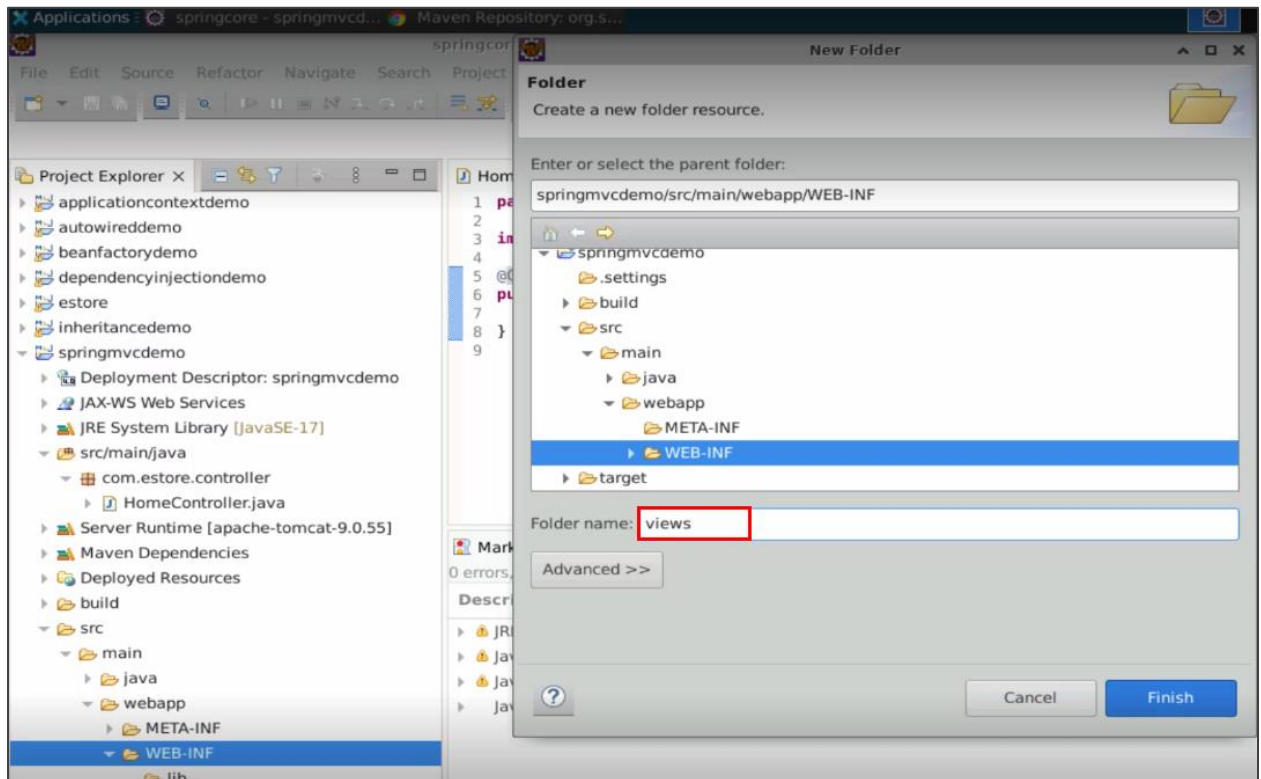


#### 4.2 Create a new folder by right-clicking on the **WEB-INF** folder and selecting **New > Folder**



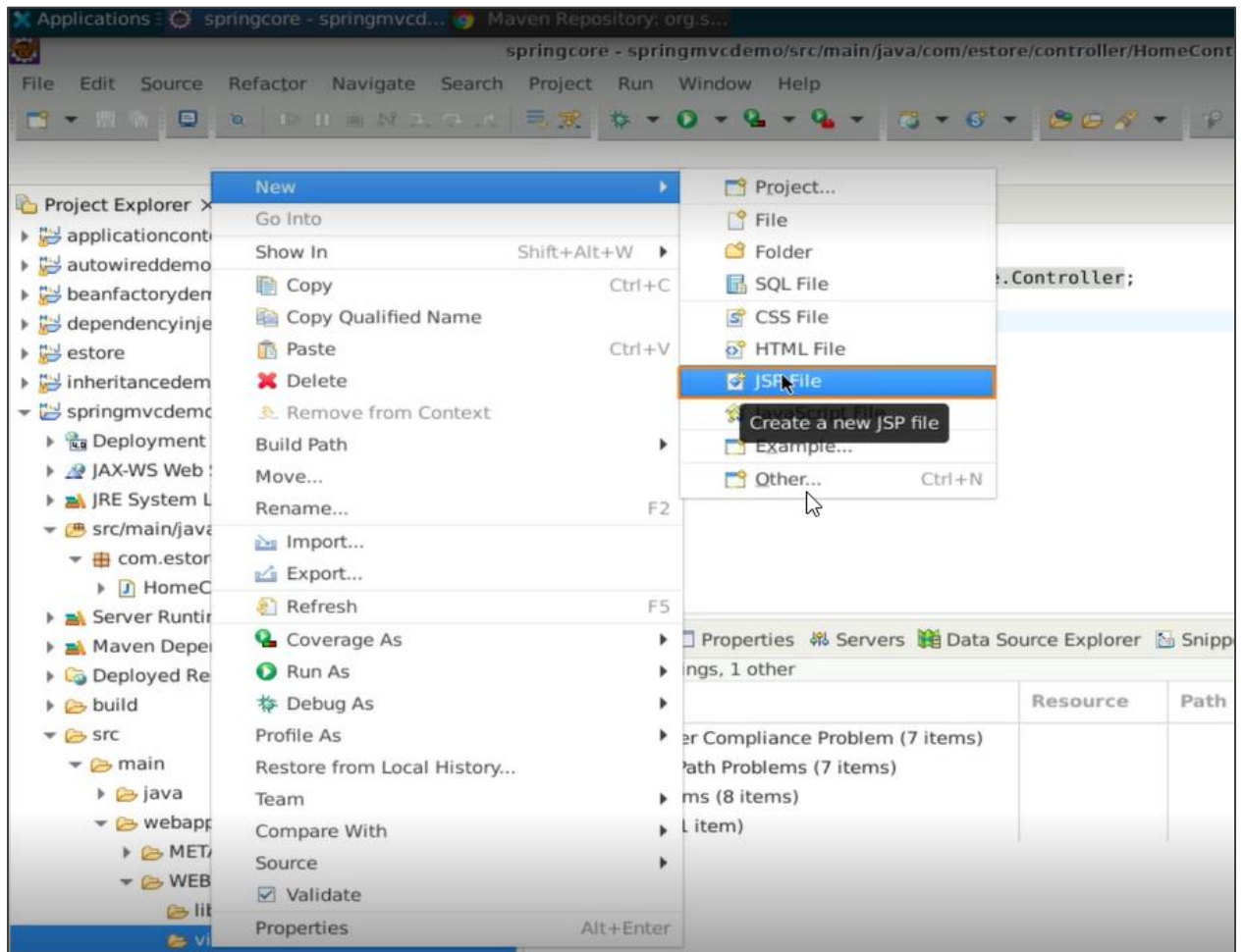


#### 4.3 Name the folder as **views** and click **Finish**

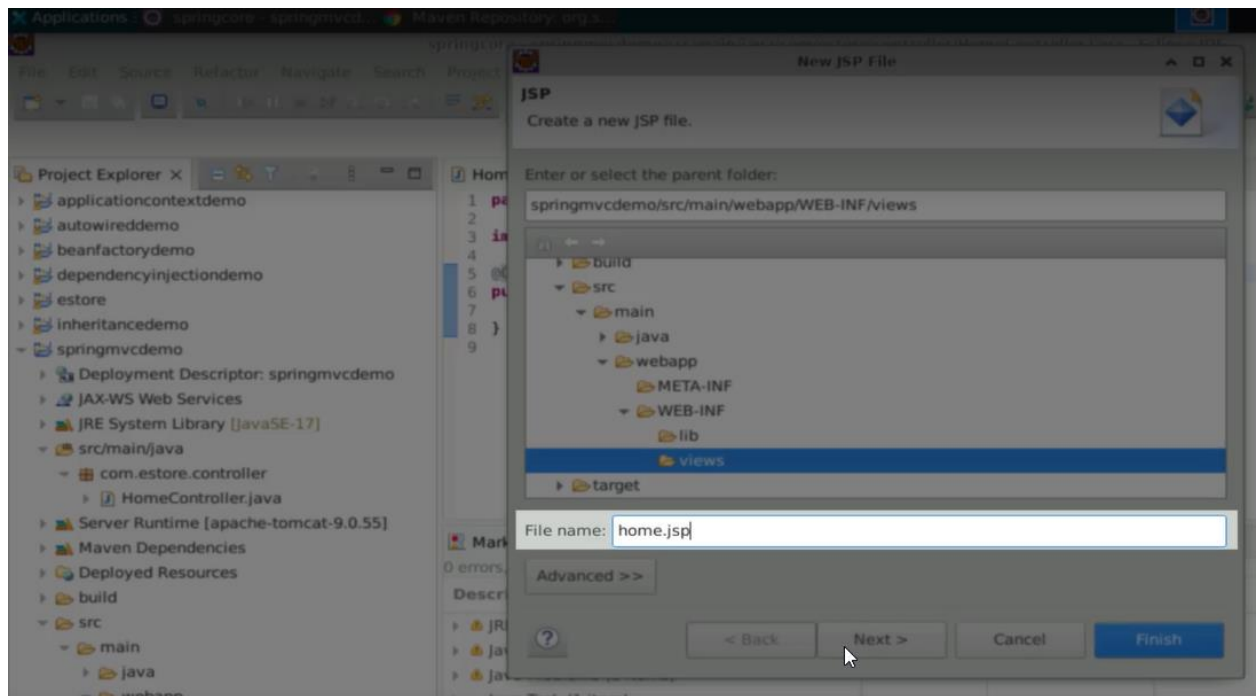




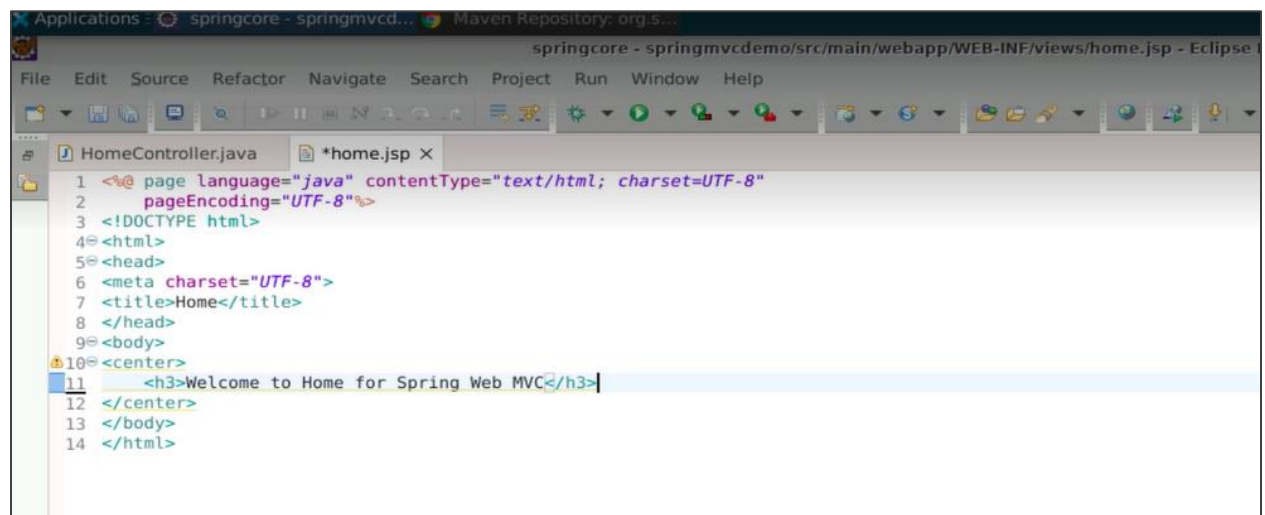
4.4 Inside the **views** folder, create a new JSP file by right-clicking on the folder and selecting **New > JSP File**



4.5 Name the file as **home.jsp** and click **Finish**

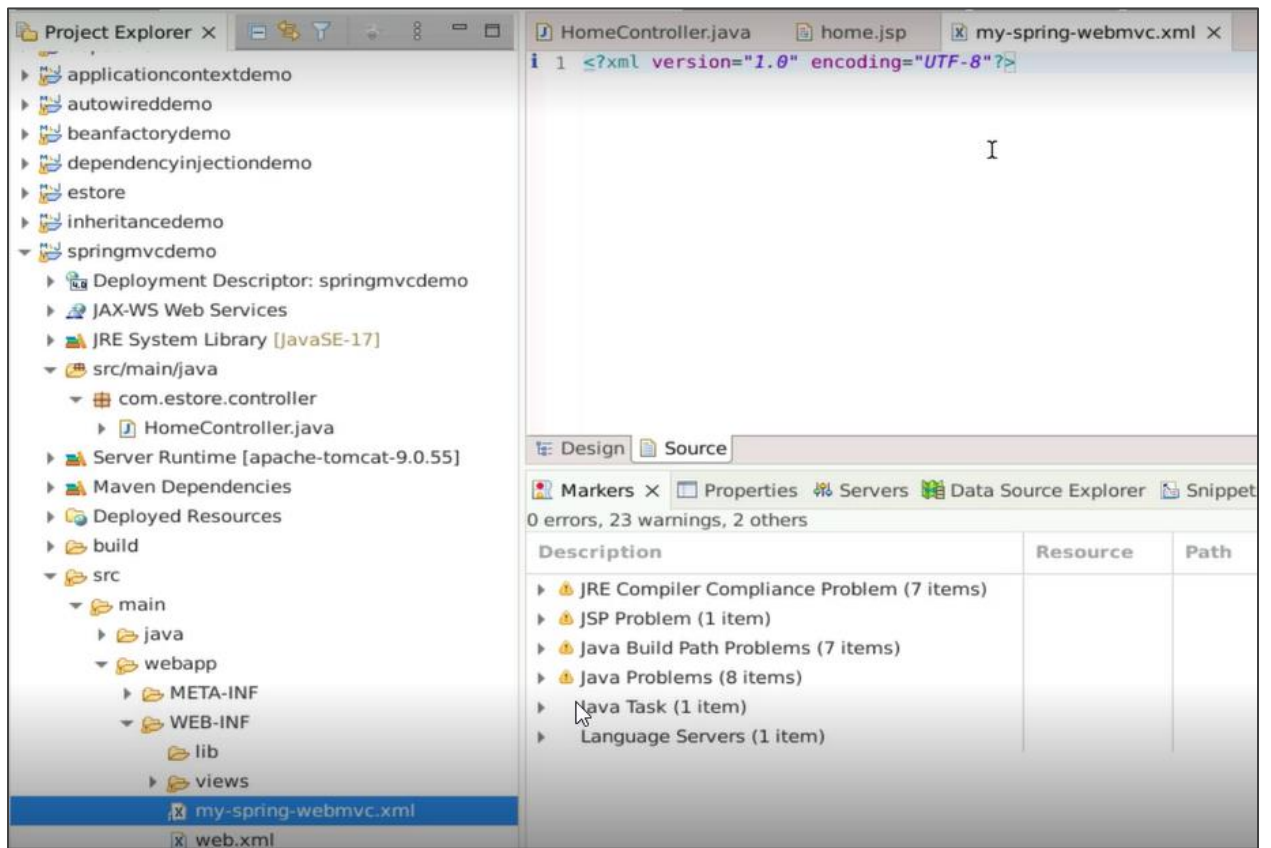


4.6 In the **home.jsp** file, update the title to **Home**. Add the text **Welcome to Home for Spring Web MVC** using an **h3** tag under the **center** tag



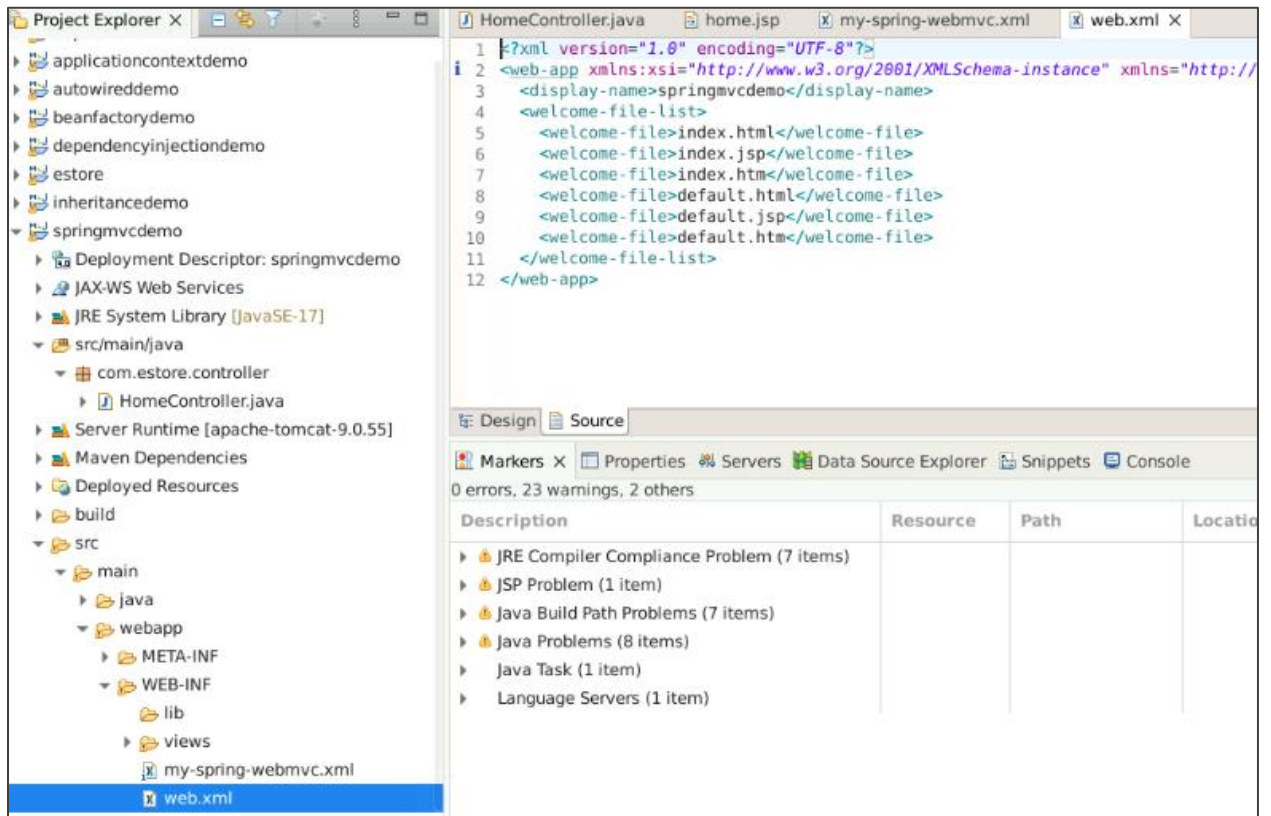
## Step 5: Configuring the DispatcherServlet

5.1 In the **webapp/WEB-INF** folder, create an XML configuration file, such as **my-spring-webmvc.xml**

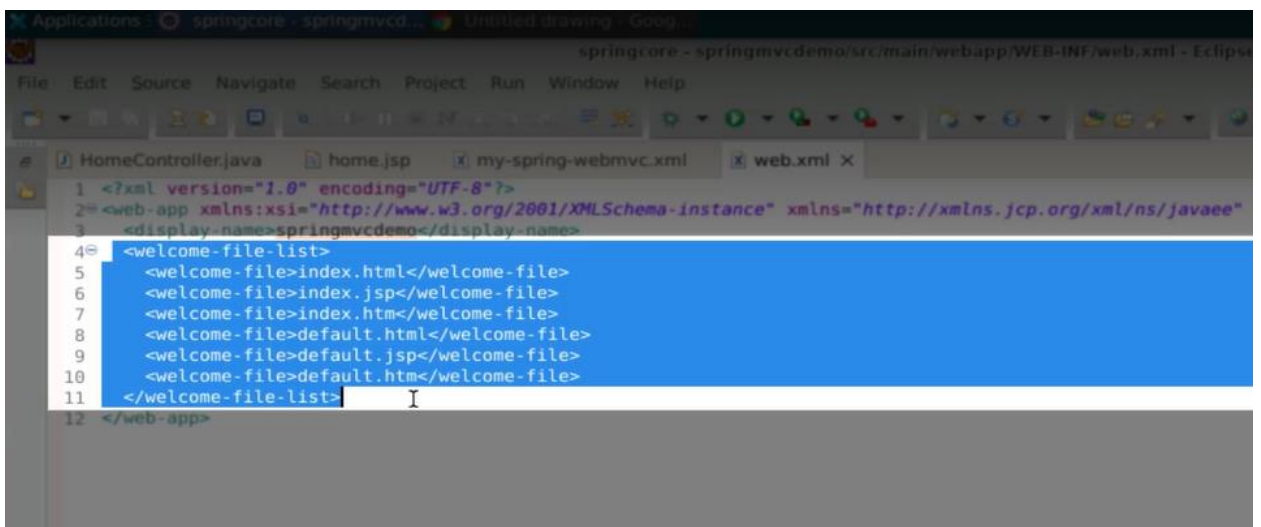


This file is where the controllers and views are configured.

## 5.2 Open the **web.xml** file located in the **WEB-INF** directory



## 5.3 Remove the existing welcome file list entries



5.4 Add a new servlet entry inside the **web-app** element



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xmlns-com
3   <display-name>springmvcdemo</display-name>
4   <servlet>
5
6   </servlet>
7 </web-app>

```

5.5 Give it a unique name, such as **FrontControllerServlet**

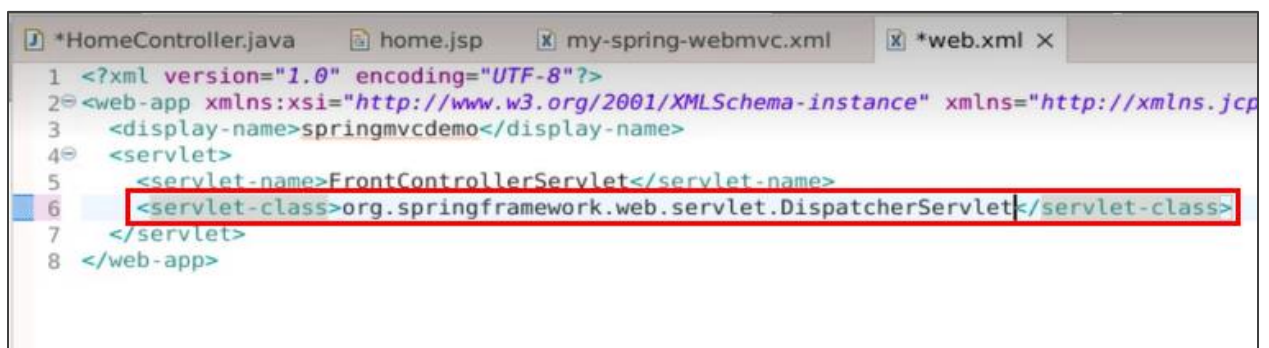


```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xmlns-com
3   <display-name>springmvcdemo</display-name>
4   <servlet>
5     <servlet-name>FrontControllerServlet</servlet-name>
6   </servlet>
7 </web-app>

```

5.6 Set the servlet class as **org.springframework.web.servlet.DispatcherServlet**



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xmlns-com
3   <display-name>springmvcdemo</display-name>
4   <servlet>
5     <servlet-name>FrontControllerServlet</servlet-name>
6     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
7   </servlet>
8 </web-app>

```

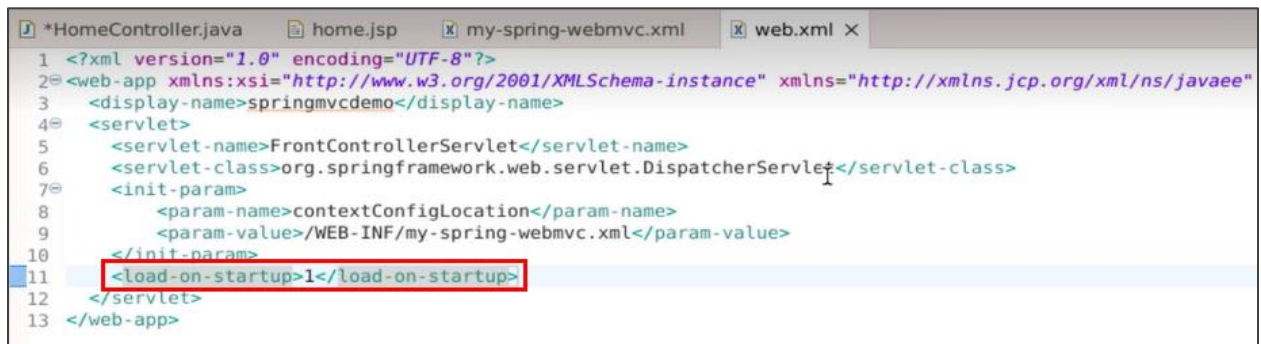


5.7 Add an initialization parameter named **contextConfigLocation** with the value pointing to the XML configuration file created earlier **my-spring-web-mvc.xml**



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3   <display-name>springmvcdemo</display-name>
4   <servlet>
5     <servlet-name>FrontControllerServlet</servlet-name>
6     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
7     <init-param>
8       <param-name>contextConfigLocation</param-name>
9       <param-value>/WEB-INF/my-spring-webmvc.xml</param-value>
10    </init-param>
11  </servlet>
12 </web-app>
```

5.8 Set the load-on-startup value to **1** to load the servlet on application startup



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3   <display-name>springmvcdemo</display-name>
4   <servlet>
5     <servlet-name>FrontControllerServlet</servlet-name>
6     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
7     <init-param>
8       <param-name>contextConfigLocation</param-name>
9       <param-value>/WEB-INF/my-spring-webmvc.xml</param-value>
10    </init-param>
11    <load-on-startup>1</load-on-startup>
12  </servlet>
13 </web-app>
```

## 5.9 Add a servlet mapping for the DispatcherServlet with the URL pattern as /

A screenshot of the Eclipse IDE showing the configuration of a web application in the web.xml file. The file is named 'web.xml' and is located in the 'WEB-INF' directory. The configuration includes a servlet named 'FrontControllerServlet' which is an instance of 'org.springframework.web.servlet.DispatcherServlet'. The servlet is configured with an init-param for 'contextConfigLocation' pointing to 'my-spring-webmvc.xml' and a load-on-startup value of 1. A servlet-mapping is also defined for the 'FrontControllerServlet' with an empty url-pattern, which is highlighted by a red rectangle in the image.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3   <display-name>springmvcdemo</display-name>
4   <servlet>
5     <servlet-name>FrontControllerServlet</servlet-name>
6     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
7     <init-param>
8       <param-name>contextConfigLocation</param-name>
9       <param-value>/WEB-INF/my-spring-webmvc.xml</param-value>
10    </init-param>
11    <load-on-startup>1</load-on-startup>
12  </servlet>
13
14  <servlet-mapping>
15    <servlet-name>FrontControllerServlet</servlet-name>
16    <url-pattern>/</url-pattern>
17  </servlet-mapping>
18 </web-app>
```

In conclusion, we successfully created a Spring MVC web project structure in Eclipse, configured the necessary components, and set up the DispatcherServlet to handle requests.