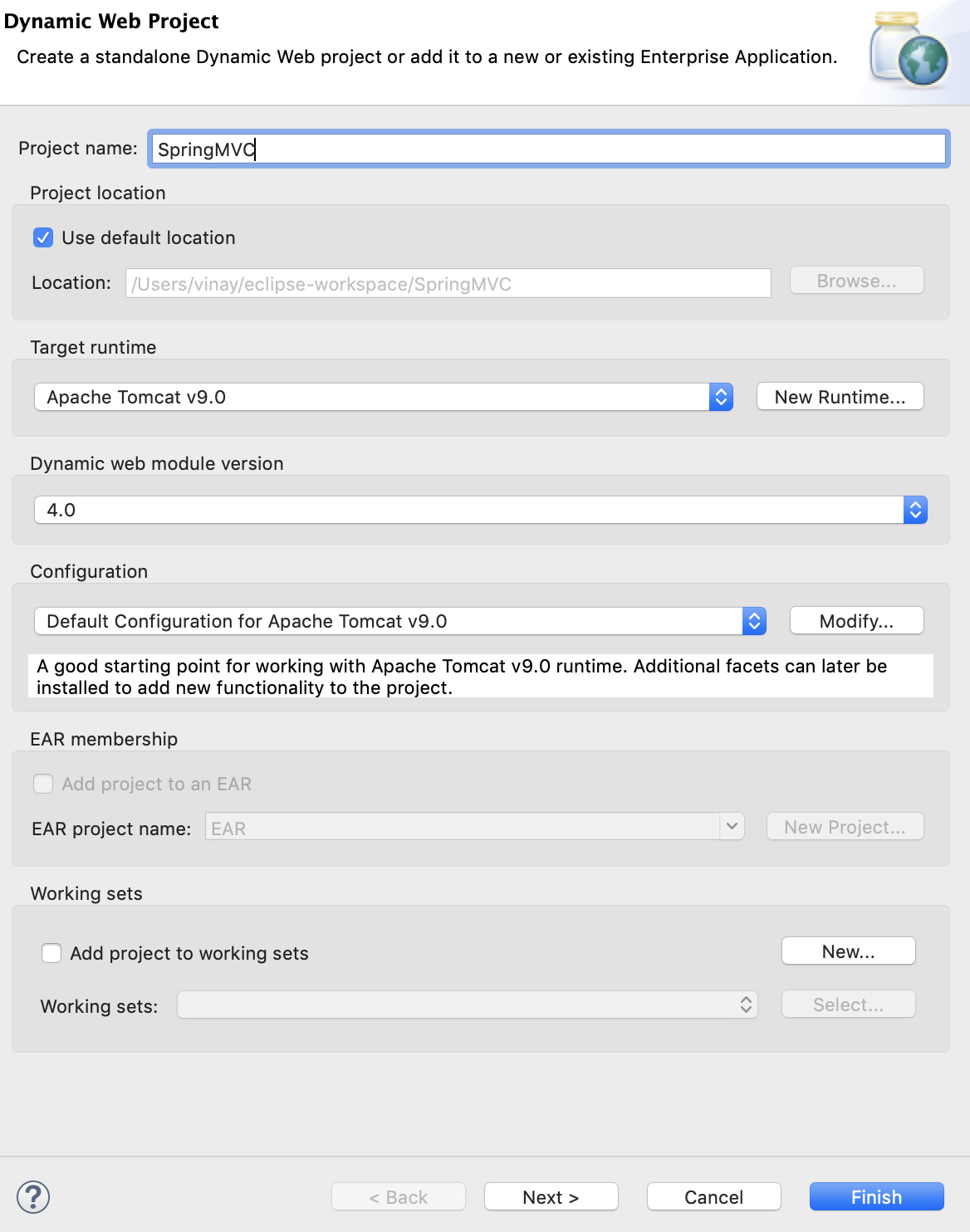
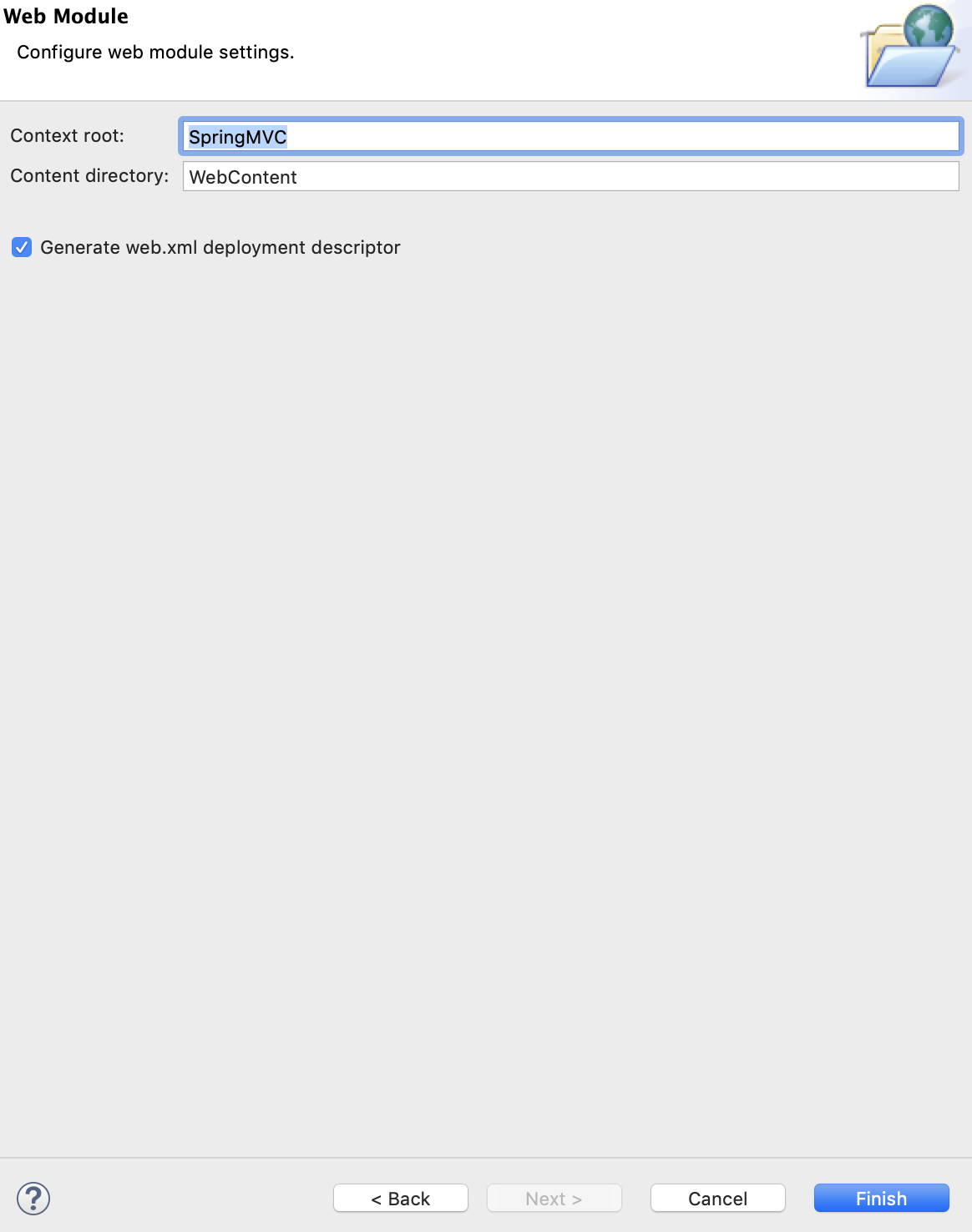
**Lab: Creating a Spring MVC application**

1. Create a dynamic web project

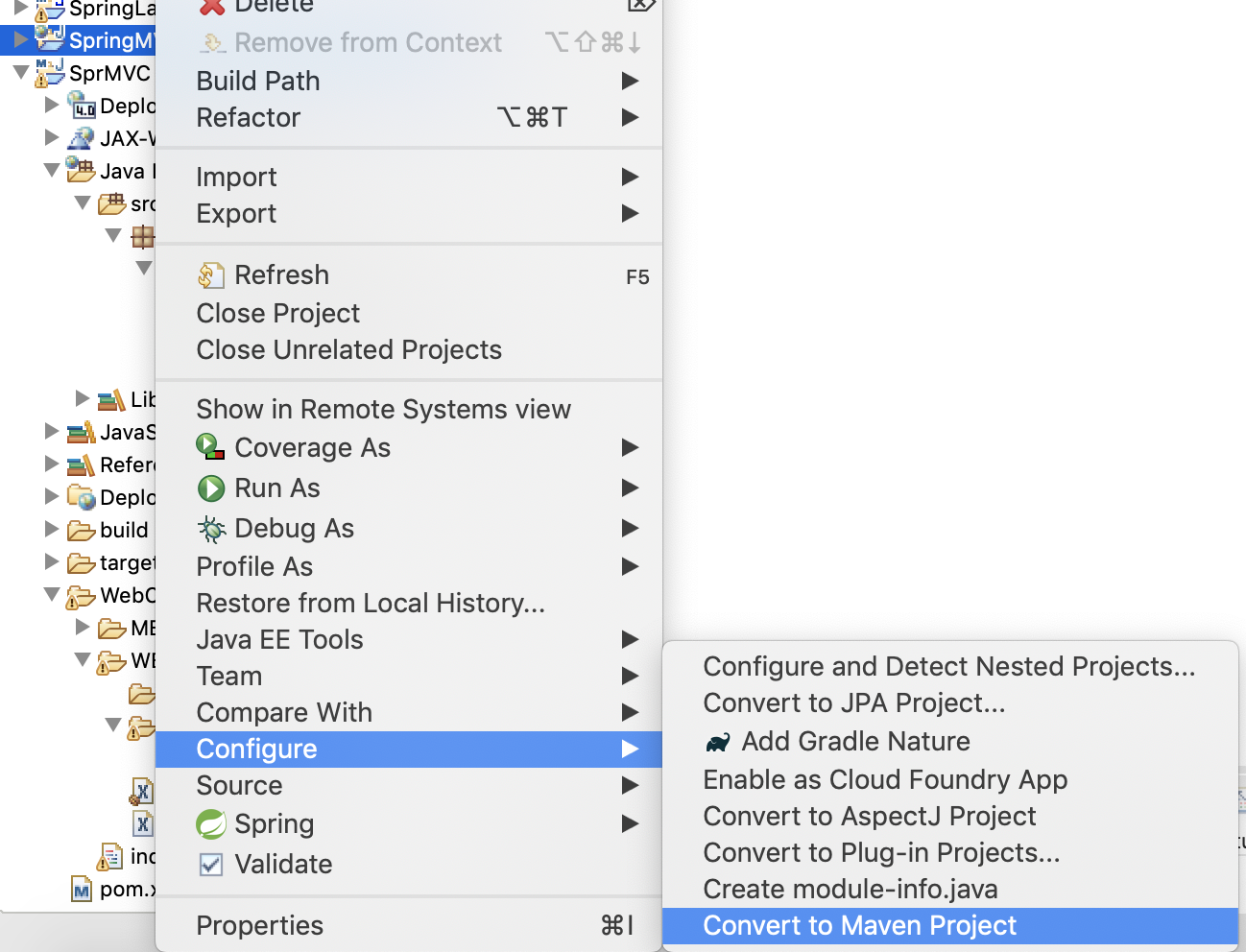


1. Select web module to get web.xml file for configuration

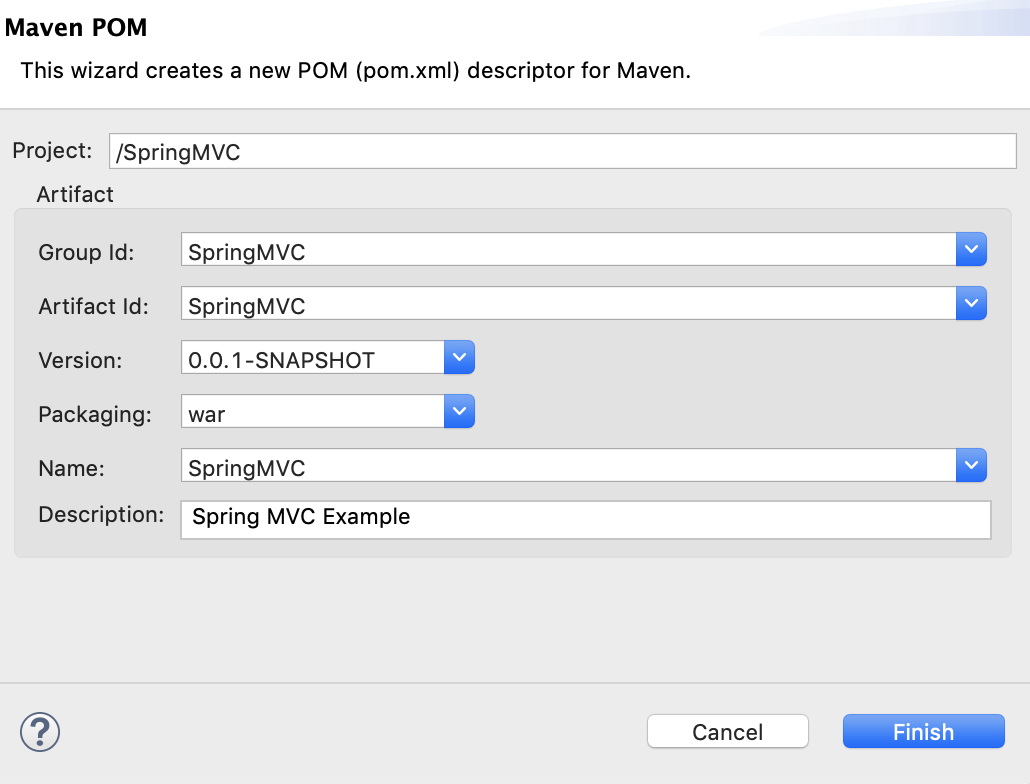


then click Finish.

1. Now right click on Project and Configure -> Convert to Maven Project.



1. Give name and description as given below



1. Open POM.xml file and include this code

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!-- Spring dependencies -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.1.9.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>5.1.9.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.1.9.RELEASE</version>

</dependency>

</dependencies>



1. Create a xml file named “dispatcher-servlet.xml” under WEB-INF folder with the following code

<servlet>beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<context:component-scan base-package=*"com.spring.mvc"* />

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"*>

<value>/WEB-INF/views/</value>

</property>

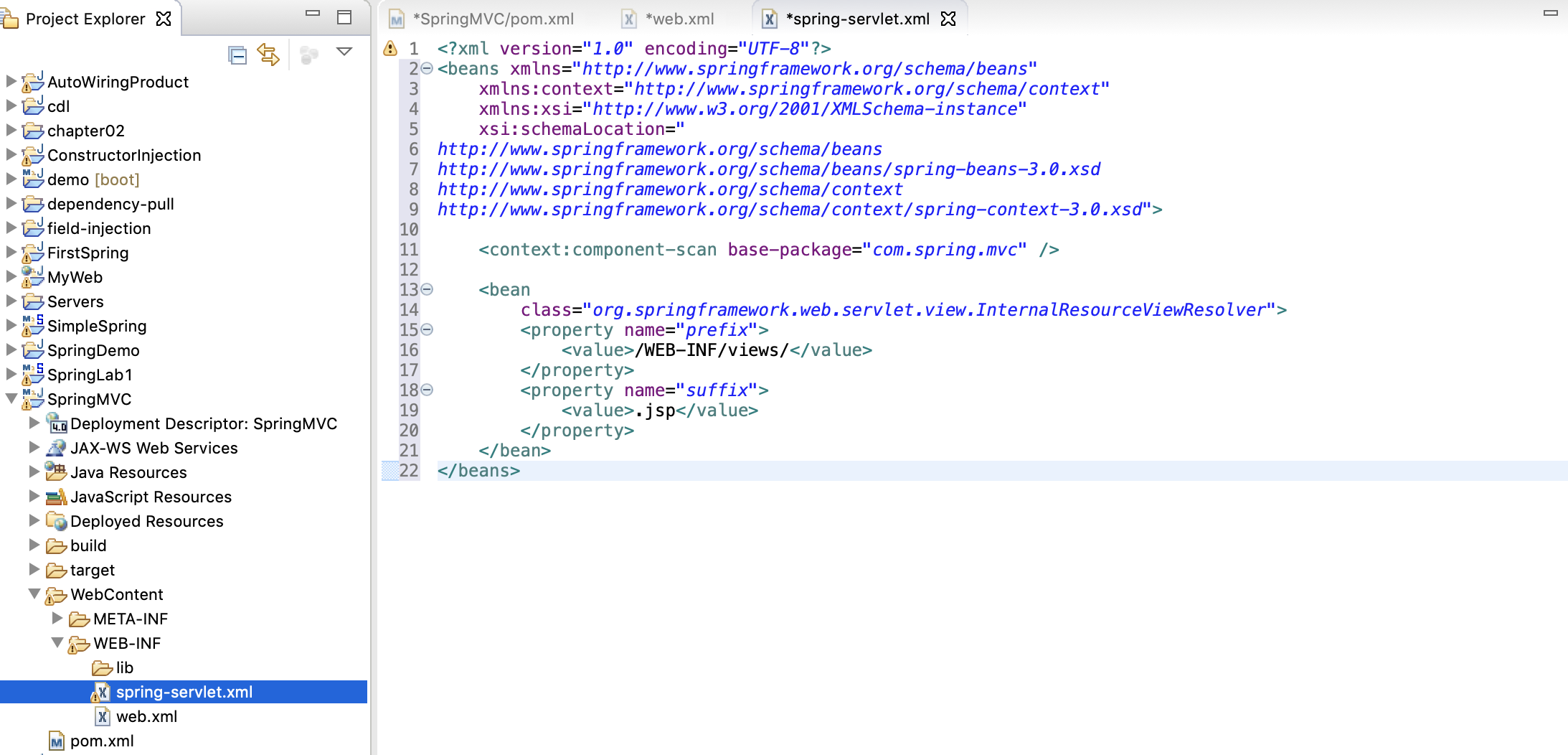
<property name=*"suffix"*>

<value>.jsp</value>

</property>

</bean>

</beans>



1. Add this code into web.xml file

<servlet-name>dispatcher</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/dispatcher-servlet.xml</param-value>

</context-param>

<listener>

<listener-class>

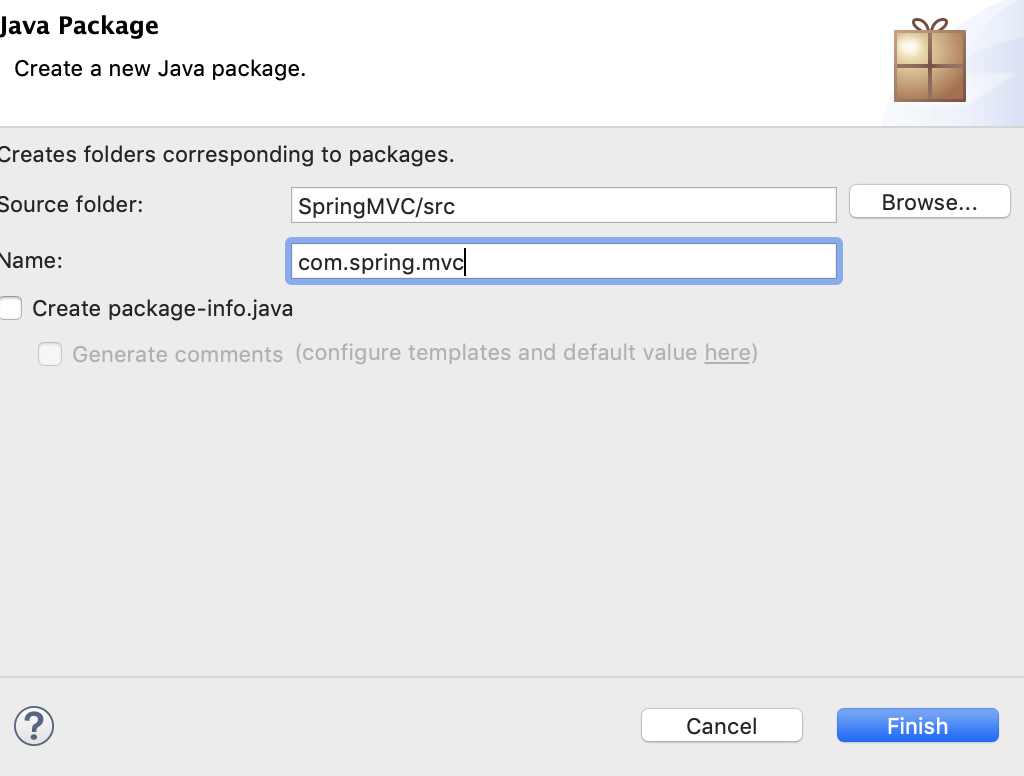
org.springframework.web.context.ContextLoaderListener

</listener-class>

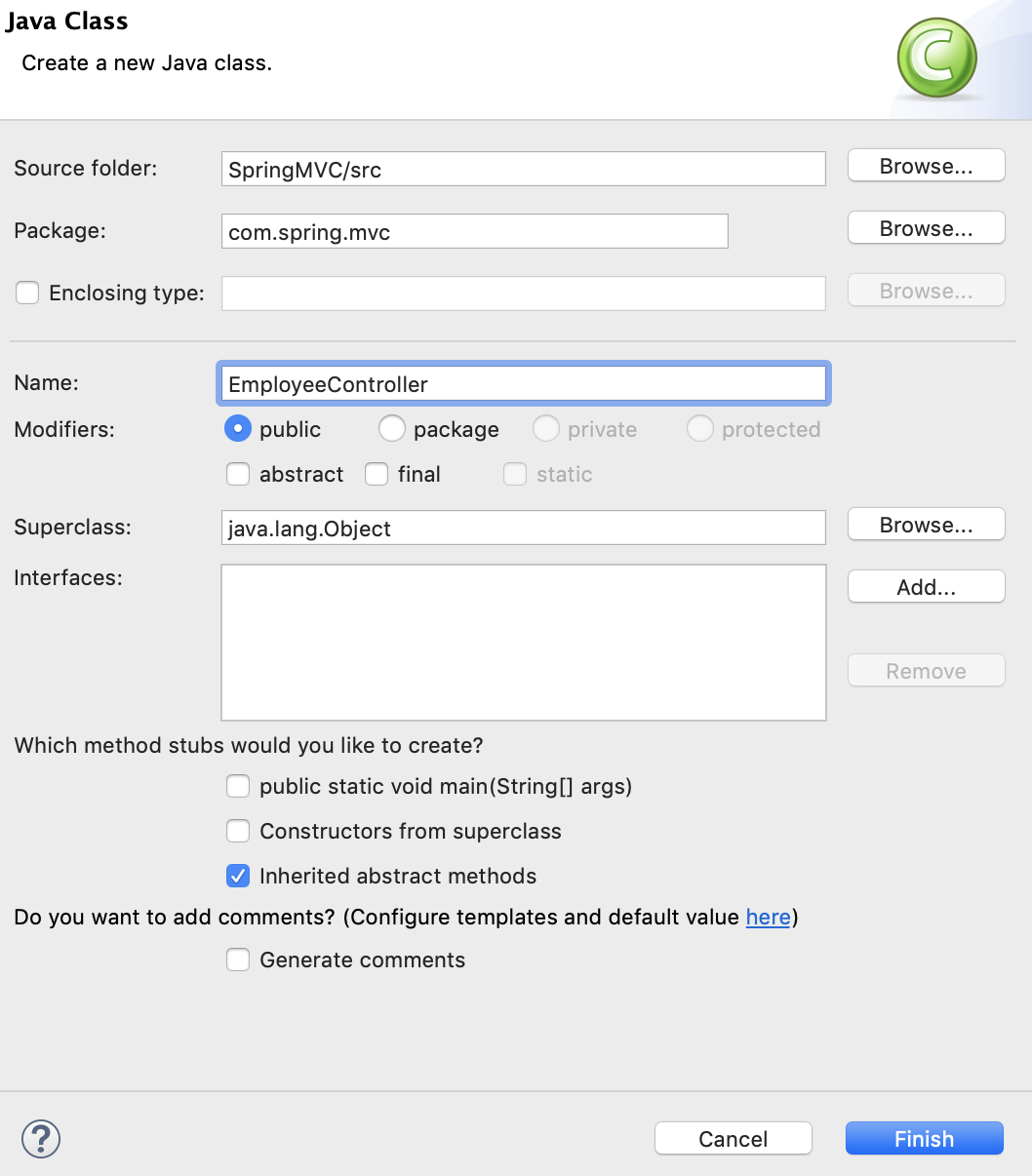
</listener>



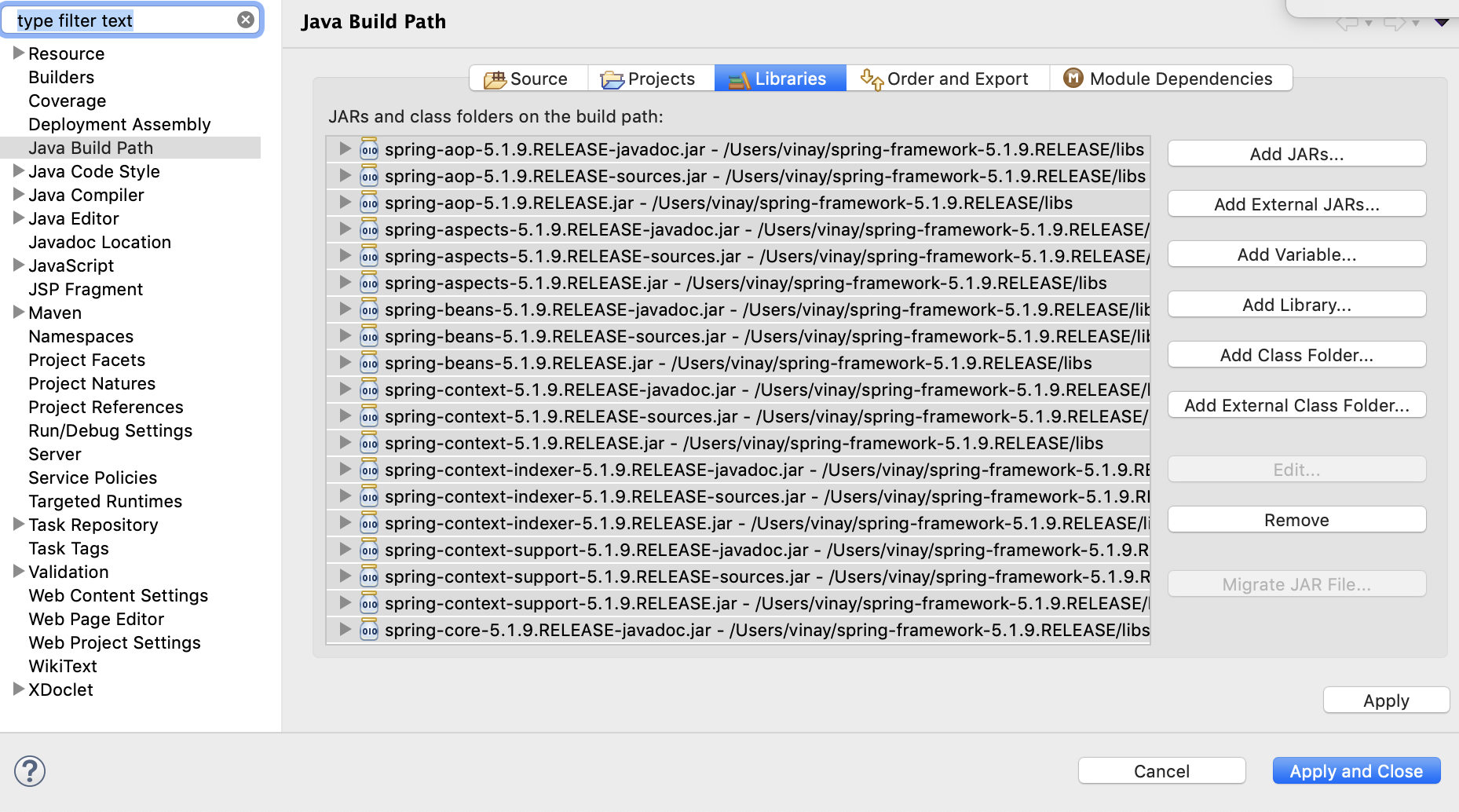
1. Create a package “com.spring.mvc” under src folder



1. Define an EmployeeController java class



Add spring package into SpringMVC project



MVC Example

**package** com.spring.mvc;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.servlet.ModelAndView;

@Controller

**public** **class** EmployeeController {

String strMessage = "Welcome to Spring MVC";

@RequestMapping("/employee")

**public** ModelAndView printMessage(

@RequestParam(value = "name", required = **false**, defaultValue = "Employee") String name) {

ModelAndView view = **new** ModelAndView("employeeView");

view.addObject("Message", strMessage);

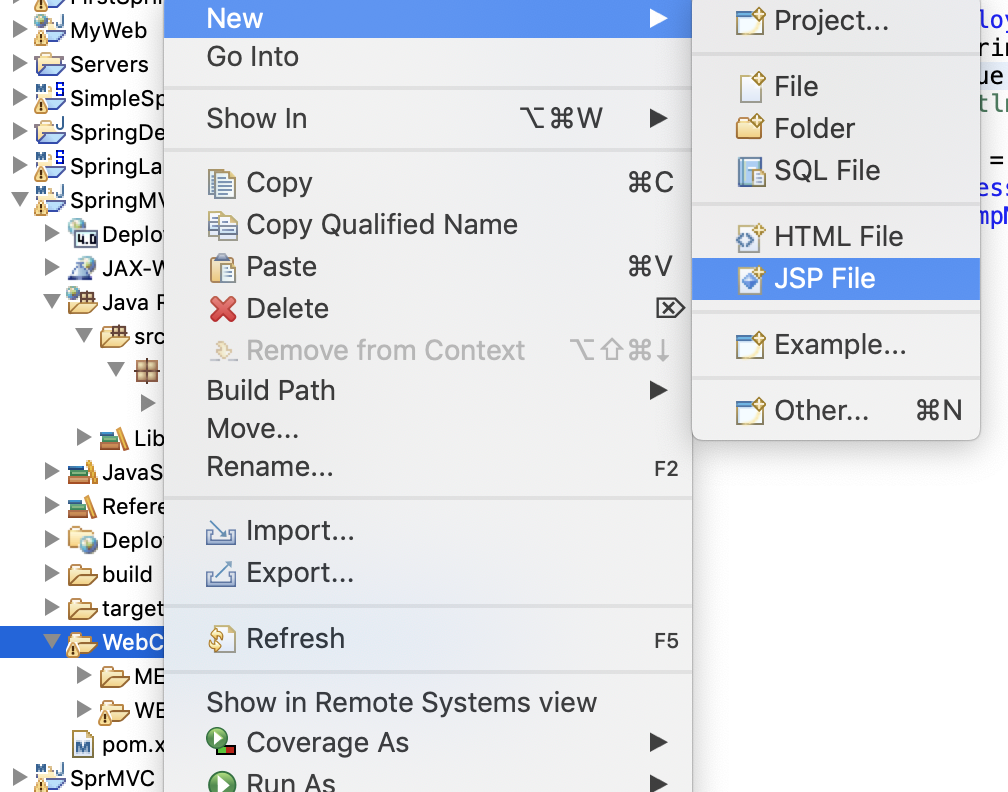
view.addObject("empName", name);

**return** view;

}

}

1. Define an index.jsp file



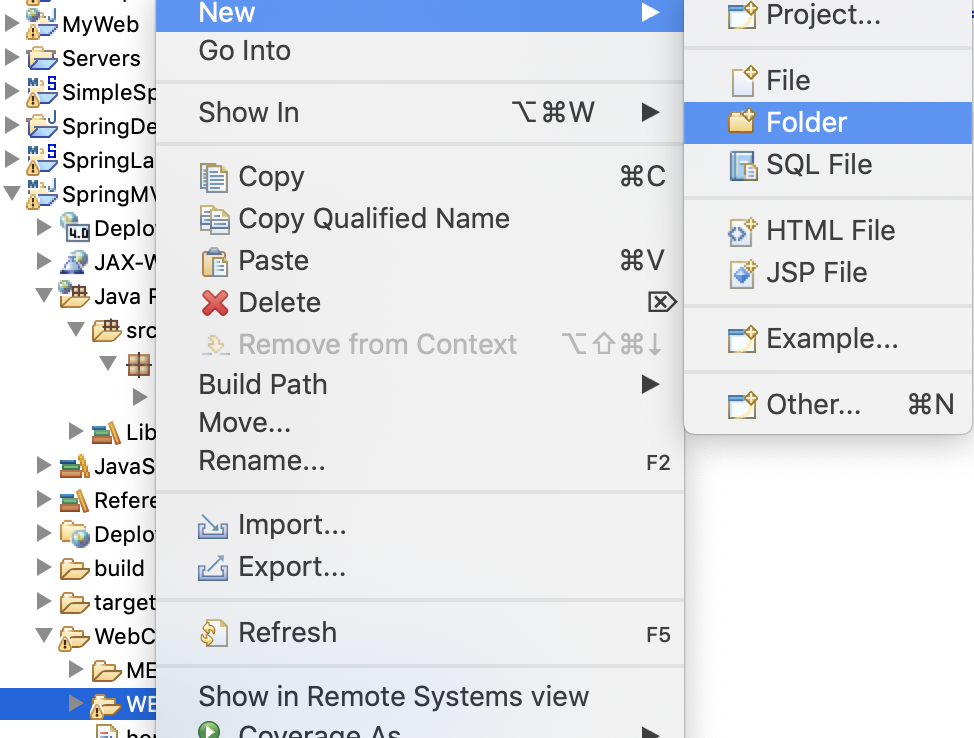
Add this code

<h2>

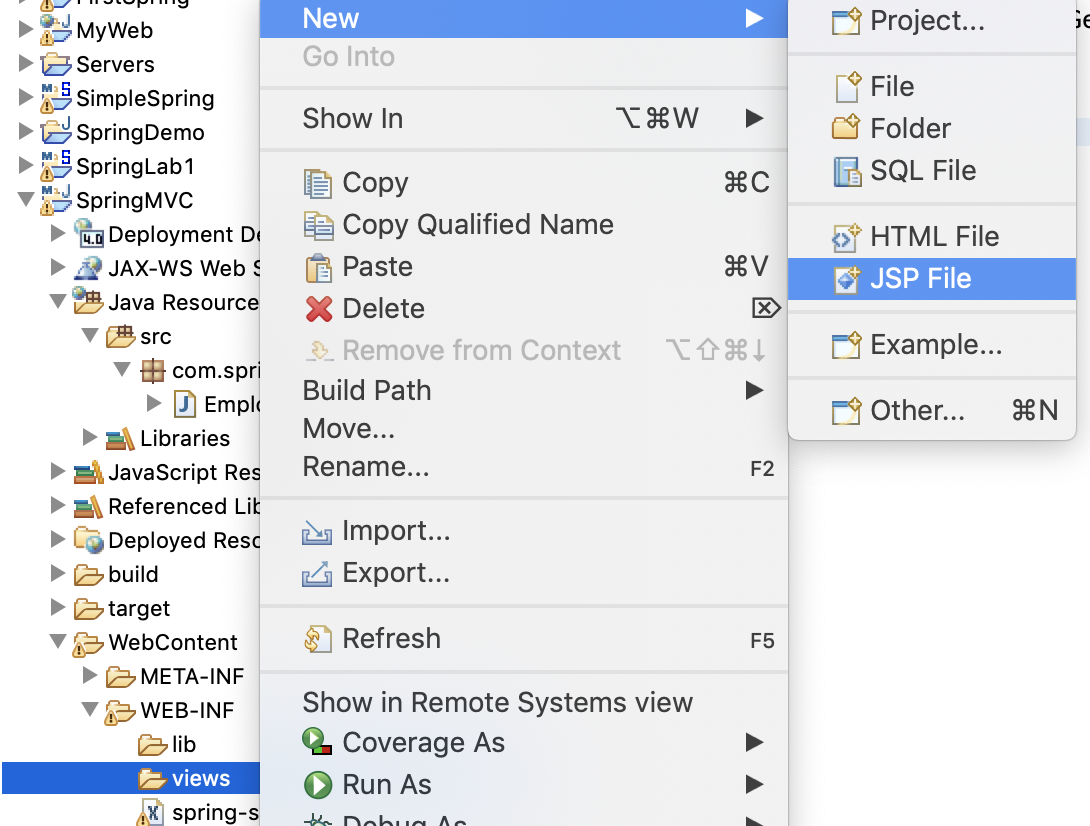
<a href=”employee?name=Max”> Get Welcome Message </a>

</h2>

1. Create a “views” folder under WEB-INF



1. Create a employeeview.jsp file under “views” folder



Add this code:

<title>Employee Spring MVC</title>

</head>

<body>

<h1> Employee Spring MVC </h1>

<p> ${Message} ${empName} </p>

</body>

1. Run this project on Server.