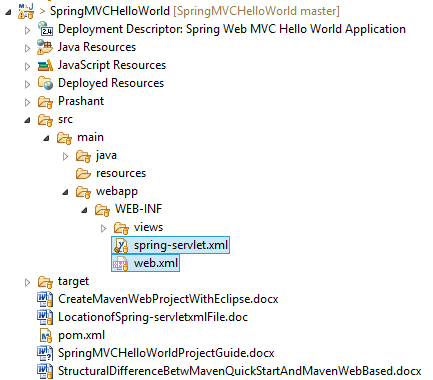
**What this project does??**

This project explains basic functioning of Spring MVC. How controller acts and handle request and render the specified jsp.

This is web based maven project.

**Note that this is xml based spring hibernate project. We have web config file web.xml ans spring configuration file spring-servlet.xml. (Location of these files are shown in screen shot of project structure below.)**

****

**Steps to create project:-**

1. This is web based maven project. Create webapp maven project. Please refer the file 'CreateWebBasedMavenProjectInEclipse.docx' in this project to know more about creation of web based maven project.

2. Add require dependencies for spring as shown in pom.xml file.

3. Create required packages and add the files.

**Functioning of the project**

When we start project, welcome file viz. welcome.jsp get rendered.

When we click 'Proceed' button it leads to rendering of page 'index.jsp'.

Now though it looks simple (and it is) we have to pay some attention to various files to understand how the flow of request takes place. These files are as follows:

1. web.xml where we have mentioned the spring functionality as DispatcherServlet.

2. spring-servlet.xml where we have defined bean 'org.springframework.web.servlet.view.InternalResourceViewResolver' which is responsible for rendering specified page mentioned in controller file.

3. com.javaLive.controller.HelloController.java is controller file of this application. Please go through annotation @RequstMapping.

To know more about @RequestMapping annotation please refer following link: <https://dzone.com/articles/using-the-spring-requestmapping-annotation>

**Note that we have action 'index' mentioned in submit button of 'welcome.jsp' page and same is there in controllers @RequestMapping which is then returns string 'index' thus leads to rendering 'index.jsp' page. As this is simple project just for understanding of Spring MVC, it does not contains any call to business logic methods otherwise it contains call to service methods which contains business logic.**

We are going to study this functionality in upcoming projects viz. 'SpringMVCArithmeticOperationsDemo' (without database operation) and 'SpringHibernateExample' (with database operations) and so on.

So please go through this project and try to understand flow of request in Spring MVC very neatly.