Spring-MVC

1. **What is MVC ?**

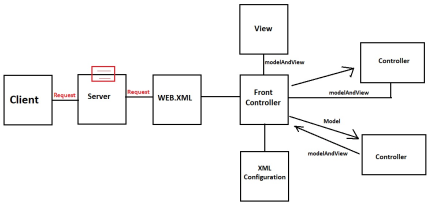
    ▼ Ans

    - It is a design pattern which is used for the development of Web Application
        - Example : Spring MVC

2. **What is Spring MVC ?**

    ▼ Ans

    - It is Spring framework which is used for the development of Web Application.
    - Spring MVC will have all the feature of Spring Core Framework.
    - Using Spring MVC, We can simplify the Web Application development. here model, view and controller are separated
    - Model → It is the data of an application, which need to be processed or persisted.
    - View → View is nothing the presentation of an Application.
    - Controller → It is used to write the business logic.
    - Front Controller → This is a controller which will intercept all the request from the client and then forward the request to the specific controller based on the request. In Spring MVC, DispatcherServlet acts a Front Controller.



45. **@Controller ?**

    ▼ Ans

    - It is a class level annotation present in org.springframework.stereotype package.
    - It is used to mark the class as controller which has to be managed by front controller (DispatcherServlet)

46. **ModelandView ?**

    ▼ Ans

    - It is a class present in org.springframework.web.servlet package.
    - It is used to return the view as well as model to the controller

    Important methods of ModelAndView

    1. setViewName(String)
        - This method is used to set the view name.
    2. addObject(String , Object)
        - This method is used to add an object into the model and view.

47. **@RequestMapping ?**

    ▼ Ans

    - **It is the annotation present in org.springframe.work.web.bind.annotation package.**
    - **It can be used for both class as well as method.**
    - **This annotation is used to map the web request to method which are prersent in RequestHandling class (Controller)**

48. **@RequestPara ?**

    ▼ Ans

    - **It is an annotation present in org.springframework.web.bind.annotation package.**
    - **This annotation can be used for a parameter.**
    - **This annotation is used to indicate that a method parameter should bound to a web request parameter.**

49. **What is JSTL (JSP Standard Tag Library) ?**

    ▼ Ans

    - **It is a library which can be used in JSP to perform some advance functionality.**

      Example : We can use if statement forEach loop, catch block etc. in the JSP using JSTL Library.

50. **Important JSTL Tags ?**

    ▼ Ans

    1. **ForEach Loop**
        - **<c:forEach var="num" items="<%=nums%>"></c:forEach>**
    2. **If Statment**
        - **<c:if test=""></c:if>**
    3. **If else Statments**
        - **<c:choose>**
            - **<c:when test=""></c:when>**
            - **<c:otherwise></c:otherwise>**
        - **</c:choose>**

51. **Code to write ForEach Loop and If Statements in JSP Using JSTL ?**

    ▼ Ans

    ▼ IfStatment.jsp

```
<%@page import="java.util.Arrays"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.util.List"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <%
    List<Integer> nums = new ArrayList<Integer>(Arrays.asList(1, 2, 3, 4, 5, 6, 8, 9));
  %>
  <c:forEach var="num" items="<%=nums%>">
```

```
    <c:if test="${num%2!=0 }">
      <h3>${num }</h3>
    </c:if>
  </c:forEach>
</body>
</html>
```

▼ **IfelseStatmemnt.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <%
    int num = 46;
  %>
  <c:choose>
    <c:when test="<%=num % 2 == 0%>">
      <h1><%=num%> is Even</h1>
    </c:when>
    <c:otherwise>
      <h1><%=num%> is Odd</h1>
    </c:otherwise>
  </c:choose>
</body>
</html>
```

52. **Steps to Create Spring MVC Project ?**

▼ **Ans**

1. **Create a Maven Project by Selecting the following Archtype.**
    a. Group **Id** → **“org.apache.maven.archetypes”**
    b. **Artifact Id** → **“maven-archetype-webapp”**
    c. **Version** → **“1.4”** (Up-to this step is enough to get version 1.4, If you doesn’t get, then adda remote catalog by below Repository URL Link)
    d. **Repository URL** → **“https://repo.maven.apache.org/maven2/archetype-catalog.xml”**
2. Group **Id** → **“org.jsp.MVCDemo” and Artifact Id** → **“Spring-MVC-Demo”**
3. **Add the following dependencies in pom.xml**
    a. **Spring Context**
    b. **Spring Web and many more (for all dependencies use Satish sir git-hub (filename is** pom.xml(Spring-MVC).txt**) or use link** → **“**https://github.com/sathishnyadav/supporting-files/blob/master/pom.xml(Spring-MVC).txt**”). copy past all the dependencies present in above link.**

▼ **pom.xml**

```
<?xml version="1.0" encoding="UTF-8"?>

<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">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.jsp.MVCDemo</groupId>
  <artifactId>Spring-MVC-Demo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>Spring-MVC-Demo Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>


    <dependency>
      <groupId>javax.servlet.jsp</groupId>
      <artifactId>jsp-api</artifactId>
      <version>2.2</version>
      <scope>provided</scope>
    </dependency>


    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>


    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.3.13</version>
    </dependency>

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.3.13</version>
    </dependency>
  </dependencies>

  <build>
    <finalName>Spring-MVC-Demo</finalName>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
      <plugins>
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
        <!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin_bindings_for_war_packaging --
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.22.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-war-plugin</artifactId>
          <version>3.2.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-install-plugin</artifactId>
          <version>2.5.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-deploy-plugin</artifactId>
          <version>2.8.2</version>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>
</project>
```

4. **Open web.xml and map the DispatcherServlet with the URL.** (do this manually or use satish sir github (File name is **web.xml(Spring-MVC).xml**) or use link or → “https://github.com/sathishnyadav/supporting-files/blob/master/web.xml(Spring-MVC).xml”)

   ▼ **web.xml**

```
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>
    <servlet>
    <servlet-name>MyMvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>MyMvc</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>
```

5. **Create an XML Configuration file with the name “DispatcherServletName-servlet.xml”. It should be present inside WEB-INF folder (DispatcherServletName should be same as configuration made web.xml). We need to configure our controller beans, for this we will get in Satish sir git-hub (filename is MyMvc-servlet.xml) or use this link → “https://github.com/sathishnyadav/supporting-files/blob/master/MyMvc-servlet.xml”**

   ▼ **MyMvc-servlet.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans.xsd
            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-context-3.0.xsd">
<context:component-scan base-package="org.jsp"></context:component-scan>
<context:annotation-config></context:annotation-config>
</beans>
```

6. **Create jsp files in webapp**

   ▼ **index.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <h3><a href="home">Click here to visit the home page</a></h3>
</body>
</html>
```

   ▼ **home.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <h1>WelCome Mr.${user }</h1>

  <h2><a href="sum.jsp">Click here to find the SUM</a></h2>
  <h2><a href="diff.jsp">Click here to find the Difference</a></h2>
  <h2><a href="prod.jsp">Click here to find the Product</a></h2>
  <h2><a href="div.jsp">Click here to find the Division</a></h2>
  <h2><a href="display">Click here to Display the data</a></h2>
</body>
</html>
```

▼ sum.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="sum">
  <input type="number" name="n1"> <br>
  <input type="number" name="n2"> <br>
  <input type="submit" name="FindSum">
  </form>
</body>
</html>
```

▼ prod.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="prod">
  <input type="number" name="n1"> <br>
  <input type="number" name="n2"> <br>
  <input type="submit" name="FindProd">
  </form>
</body>
</html>
```

▼ div.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form action="div">
  <input type="number" name="n1"> <br>
  <input type="number" name="n2"> <br>
  <input type="submit" name="FindDiv">
  </form>
</body>
</html>
```

▼ **diff.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form action="diff">
  <input type="number" name="n1"> <br>
  <input type="number" name="n2"> <br>
  <input type="submit" name="FindDiff">
  </form>
</body>
</html>
```

▼ **printResult.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h2>${result }</h2>
</body>
</html>
```

▼ **ViewData.jsp (For this, We need to import JSTL tag. Link is getting in Satish Sir git-hub (File name tags.txt)) or link → "https://github.com/sathishnyadav/supporting-files/blob/master/tags.txt"**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@page import="org.jsp.controller.User"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<c:forEach var="us" items="${us }">
<h2>${us.getId() }    ${us.getName() }   ${us.getSal() }</h2>
</c:forEach>
</body>
</html>
```

7. **Create a org.jsp.constroller.HomeController in src/main/java folder (If we don't get this folder, do this go to properties and set java build path)**

▼ **org.jsp.constroller**

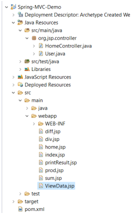
▼ **HomeController.java**

```
package org.jsp.controller;
```

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import javax.servlet.http.HttpServletRequest;
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 HomeController {
  @RequestMapping("/home")
  public ModelAndView home() {
    ModelAndView modelAndView = new ModelAndView();
    modelAndView.setViewName("home.jsp");
    modelAndView.addObject("user", new ArrayList<>(Arrays.asList("A", "B", "C")));
    return modelAndView;
  }

  @RequestMapping("/prod")
  public ModelAndView findProd(HttpServletRequest request) {
    int num1 = Integer.parseInt(request.getParameter("n1"));
    int num2 = Integer.parseInt(request.getParameter("n2"));
    String result = "The Sum is " + (num1 * num2);
    ModelAndView modelAndView = new ModelAndView();
    modelAndView.setViewName("printResult.jsp");
    modelAndView.addObject("result", result);

    return modelAndView;
  }

  // We can reduce the code like this
  @RequestMapping("/sum")
  public ModelAndView findSum(@RequestParam(name = "n1") int a, @RequestParam(name = "n2") int b,
      ModelAndView modelAndView) {
    String result = "The Sum is " + (a + b);
    modelAndView.addObject("result", result);
    modelAndView.setViewName("printResult.jsp");

    return modelAndView;
  }

  // We can again reduce the code like this
  @RequestMapping("/diff")
  public ModelAndView findDiff(int n1, int n2, ModelAndView modelAndView) {
    String result = "The Sum is " + (n1 - n2);
    modelAndView.setViewName("printResult.jsp");
    modelAndView.addObject("result", result);

    return modelAndView;
  }

  @RequestMapping("/div")
  public ModelAndView findDiv(HttpServletRequest request) {
    int num1 = Integer.parseInt(request.getParameter("n1"));
    int num2 = Integer.parseInt(request.getParameter("n2"));
    String result = "The Sum is " + (num1 / num2);
    ModelAndView modelAndView = new ModelAndView();
    modelAndView.setViewName("printResult.jsp");
    modelAndView.addObject("result", result);

    return modelAndView;
  }

  @RequestMapping("/display")
  public ModelAndView displayUser(ModelAndView view) {
    List<User> users = new ArrayList<User>();
    users.add(new User(1, 123, "Manu"));
    users.add(new User(2, 456, "OK"));
    users.add(new User(3, 789, "Bye"));
    view.setViewName("ViewData.jsp");
    view.addObject("us", users);
    return view;
  }
}
```

▼ **User.java**

```
package org.jsp.controller;

public class User {
  private int id, sal;
  private String name;

  public User(int id, int sal, String name) {
    super();
    this.id = id;
    this.sal = sal;
    this.name = name;
  }

  public User() {
  }

  public int getId() {
    return id;
  }

  public void setId(int id) {
    this.id = id;
  }

  public int getSal() {
    return sal;
  }

  public void setSal(int sal) {
    this.sal = sal;
  }

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }
}
```



```
Spring-MVC-Demo
  Deployment Descriptor: Archetype Created We
  Java Resources
    src/main/java
      org.jsp.controller
        HomeController.java
        User.java
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
    main
      java
      webapp
        WEB-INF
        diff.jsp
        div.jsp
        home.jsp
        index.jsp
        printResult.jsp
        prod.jsp
        sum.jsp
        ViewData.jsp
    test
  target
  pom.xml
```

53. **@ResponseBody ?**

    ▼ Ans

    - **It is an annotation present in org.springframework.web.bind.annotation package.**
    - **It can be used for a method or for class.**
    - **It is used to indicate that, a method returned value should be considered as a response.**

54. **Code for @Responsebody ?**

    ▼ Ans

    ▼ org.jsp.ontroller

    ▼ HomeController.java

```
package org.jsp.ontroller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
```

```
@Controller
public class HomeController {
  @RequestMapping("/hii")
  @ResponseBody
  public String hello() {
    return "Hello from HomeController";
  }
}
```

index.jsp

```
<html>
<body>
<h2><a href="hii">Say Hi to HomeController by Clicking</a></h2>
</body>
</html>
```

MyMvc-servlets.xml

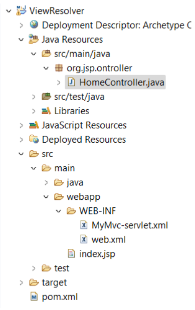
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans.xsd
            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-context-3.0.xsd">
<context:component-scan base-package="org.jsp"></context:component-scan>
<context:annotation-config></context:annotation-config>
</beans>
```

web.xml

```
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>
  <servlet>
    <servlet-name>MyMvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>MyMvc</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>
```

pom.xml

- **Add dependencies → Satish sir Git-Hub file name(pom.xml(Spring-MVC).txt)**
  “https://github.com/sathishnyadav/supporting-files/blob/master/pom.xml(Spring-MVC).txt**”**



```
ViewResolver
  Deployment Descriptor: Archetype C
  Java Resources
    src/main/java
      org.jsp.ontroller
        HomeController.java
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
    main
      java
      webapp
        WEB-INF
          MyMvc-servlet.xml
          web.xml
        index.jsp
    test
  target
  pom.xml
```

55. **What is ViewResolver ?**

    ▼ **Ans**

    - **It is an interface present in**
    - **This interface should be implemented by the classes that can resolve the views by name.**
    - **ViewResolver helps to achieve loose coupling between controller and views.**
    - **Following are the implementation classes of ViewResolver.**
        1. **InternalResourcesViewResolver. (JRVR)**
        2. **ContextNagotiatingViewResolver. (CNVR)**
        3. **BeanNameViewResolver. (BNVR)**
    - **JRVR - It is the subclass of URL based ViewResolver. We need to specify 2 properties for JRVR.**
        1. **Prefix**
        2. **Suffix**

56. **What is Session ?**

    ▼ **Ans**

    - **It is a particular interval of time state of the user will be managed. We can Session with the help of an interface HttpSession.**

57. **What is Session Management or Tracking ?**

    ▼ **Ans**

    - **Http protocol is a stateless protocol. So Every Request to the server will be consider as a new Request. We can manage the Session or track the Session of a user in following ways.**
        1. **Cookies**
        2. **Hidden Form Fields**
        3. **URL Rewriting**
        4. **HttpSession**

58. **@SessionAttribute ?**

    ▼ **Ans**

    - **It is an annotation present in org.springframework.web.bind.annotation package.**
    - **It is used to list the names of model attributes that needs to be stored in a Session.**
    - **It s a Class level Annotation.**

59. **@ModelAttribute ?**

    ▼ **Ans**

- **It is an annotation present in org.springframework.web.bind.annotation package.**
- **It can be used for a method as well as a parameter.**
- **This annotation is used to bind a method parameter with a named model attribute.**

60. **Code for ViewResolver , @SessionAttribute , @ModelAttribute ?**

▼ **Ans**

▼ **pom.xml**

- **Add dependencies → Satish sir Git-Hub file name(pom.xml(Spring-MVC).txt) "https://github.com/sathishnyadav/supporting-files/blob/master/pom.xml(Spring-MVC).txt"**

▼ **MyMvc-serlet.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans.xsd
            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-context-3.0.xsd">
  <context:component-scan base-package="org.jsp"></context:component-scan>
  <context:annotation-config></context:annotation-config>

  <bean id="viewresolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/myviews/"></property>
    <property name="suffix" value=".jsp"></property>
  </bean>
</beans>
```

▼ **web.xml**

```
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>
  <servlet>
    <servlet-name>MyMvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>MyMvc</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>
```

▼ **index.jsp**

```
<html>
<body>
  <h2>Hello World !</h2>
  <h2><a href="hii">Say Hi to HomeController by Clicking</a></h2>
  <a href="home">Home Page</a><br>
  <a href="load">Clcik here to Register</a>
</body>
</html>
```

▼ **home.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<!DOCTYPE html>
```

```
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <h2>Hello Mr.${user }</h2>
  <h2><a href="verify">Click here to Verify the Session</a></h2>
</body>
</html>
```

display.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <%
    String user = (String) session.getAttribute("user");
    String person = (String) session.getAttribute("person");
    String meet = (String) session.getAttribute("me");
  %>
  <h1>User:${user }</h1>
  <h1>Person:${person }</h1>
  <h1>me:${meet }</h1>
</body>
</html>
```

register.jsp (To use Spring Form. We need use one Library. that we will get it in Satish Sir Git-hub (File name (**tags.txt**)) or use Link → “https://github.com/sathishnyadav/supporting-files/blob/master/tags.txt” (1st Tag copy and past)

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form:form modelAttribute="user" action="reg" method="post">
    <form:label path="name">Name</form:label>
    <form:input path="name" />
    <br>
    <form:label path="age">Age</form:label>
    <form:input path="age" />
    <br>
    <form:label path="phone">Phone</form:label>
    <form:input path="phone" />
    <br>
    <form:label path="password">Password</form:label>
    <form:input path="password" />
    <br>
    <form:button>Register</form:button>
  </form:form>
</body>
</html>
```

org.jsp.controller

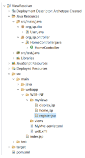
HomeController.jsp

```
package org.jsp.ontroller;

import org.jsp.dto.User;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.MatrixVariable;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.SessionAttributes;
import org.springframework.web.servlet.ModelAndView;

@Controller
@SessionAttributes(names = { "user", "person" })
public class HomeController {
  @RequestMapping("/load")
  public ModelAndView load(ModelAndView view) {
    view.setViewName("register");
    view.addObject("user", new User());
    return view;
  }

  @RequestMapping("/hii")
  @ResponseBody
  public String hello() {
    return "Hello from HomeController";
  }

  @RequestMapping("/home")
  public ModelAndView home(ModelAndView view) {
    view.setViewName("home");
    view.addObject("user", "abc");
    return view;
  }

  @RequestMapping(value = "/verify")
  public ModelAndView verifySession(ModelAndView view) {
    view.addObject("user", "abc");
    view.addObject("person", "xyz");
    view.addObject("me", "hello");
    view.setViewName("display");
    return view;
  }

  @RequestMapping(value = "/reg", method = RequestMethod.POST)
  public void register(@ModelAttribute User user) {
    System.out.println("Name : " + user.getName());
    System.out.println("Age : " + user.getAge());
    System.out.println("Phone : " + user.getPhone());
    System.out.println("Password : " + user.getPassword());
  }
}
```

org.jsp.dto

User.java

```
package org.jsp.dto;

public class User {
  private String name, password;
  private int age;
  private long phone;

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }

  public String getPassword() {
    return password;
  }

  public void setPassword(String password) {
    this.password = password;
  }

  public int getAge() {
    return age;
  }

  public void setAge(int age) {
    this.age = age;
  }

  public long getPhone() {
    return phone;
  }

  public void setPhone(long phone) {
    this.phone = phone;
  }

}
```



```
ViewResolver
  Deployment Descriptor: Archetype Created
  Java Resources
    src/main/java
      org.jsp.dto
        User.java
      org.jsp.ontroller
        HomeController.java
          HomeController
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
    main
      java
      webapp
        WEB-INF
          myviews
            display.jsp
            home.jsp
            register.jsp
          views
          MyMvc-servlet.xml
          web.xml
        index.jsp
    test
  target
  pom.xml
```

61. **Code for CRUD Operation Using Spring-MVC and Hibernate ?**

- ▼ **Ans**

  1. **Create Maven Project by selecting archtype - 1.4**
  2. **Add the dependecies to pom.xml (We will get it in Satish Sir Github)**
  3. **Web need configure the dispacherServlet with URL**
  4. **Create configuration file by name called (_ _ _Mvc-servlet.xml)**
  5. **Create packages like dto, dao, service, controller.**
  6. **Create persistence.xml in WEB-INF → classes → META-INF → persistence.xml**

  - ▼ **org.jsp.SpringHibernateApp.dto**
    - ▼ **User.java**

```
package org.jsp.SpringHibernateApp.dto;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class User {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private int id;
  private String name, password;
  private long phone;
  private int age;

  public int getId() {
    return id;
  }

  public void setId(int id) {
    this.id = id;
  }

  public int getAge() {
    return age;
  }

  public void setAge(int age) {
    this.age = age;
  }
  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }

  public String getPassword() {
    return password;
  }

  public void setPassword(String password) {
    this.password = password;
  }

  public long getPhone() {
    return phone;
  }

  public void setPhone(long phone) {
    this.phone = phone;
  }
}
```

▼ org.jsp.SpringHibernateApp.dao

▼ UserDao.java

```
package org.jsp.SpringHibernateApp.dao;

import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.EntityTransaction;
import javax.persistence.Query;

import org.jsp.SpringHibernateApp.dto.User;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

@Repository
public class UserDao {
  @Autowired
  EntityManager manager;

  public User SaveUser(User user) {
    EntityTransaction transaction = manager.getTransaction();
    manager.persist(user);
    transaction.begin();
    transaction.commit();
    return user;
  }

  public User UpdateUser(User user) {
    EntityTransaction transaction = manager.getTransaction();
    manager.merge(user);
    transaction.begin();
    transaction.commit();
    return user;
  }

  public User GetUserById(int id) {
    return manager.find(User.class, id);
  }

  public boolean DeleteUser(int id) {
    User u = manager.find(User.class, id);
    if (u != null) {
      EntityTransaction transaction = manager.getTransaction();
      manager.remove(u);
      transaction.begin();
      transaction.commit();
      return true;
    }
    return false;
  }
  public List<User> FindAllUser(){
    Query q = manager.createQuery("select u from User u");
    return q.getResultList();
  }

  public User VerifyUser(long phone, String password) {
    Query q = manager.createQuery("select u from User u where u.phone=?1 and u.password=?2");
    q.setParameter(1, phone);
    q.setParameter(2, password);
    List<User> users = q.getResultList();
    if (users.size() > 0)
      return users.get(0);
    return null;
  }
}
```

org.jsp.SpringHibernateApp.service

UserService.java

```
package org.jsp.SpringHibernateApp.service;

import java.util.List;

import org.jsp.SpringHibernateApp.dao.UserDao;
import org.jsp.SpringHibernateApp.dto.User;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class UserService {
  @Autowired
  private UserDao dao;

  public User SavaUser(User user) {
    return dao.SaveUser(user);
  }

  public User UpdateUser(User user)
  {
    return dao.UpdateUser(user);
  }

  public User GetUserById(int id)
  {
    return dao.GetUserById(id);
  }

  public boolean DeleteUser(int id)
  {
    return dao.DeleteUser(id);
  }
  public List<User> FindAllUser(){
    return dao.FindAllUser();
  }
  public User VerifyUser(long phone, String password)
  {
    return dao.VerifyUser(phone, password);
  }
}
```

org.jsp.SpringHibernateApp

UserConfig.java

```
package org.jsp.SpringHibernateApp;

import javax.persistence.EntityManager;
import javax.persistence.Persistence;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@Configuration
```

```
@ComponentScan(basePackages = "org.jsp.SpringHibernateApp")
public class UserConfig
{
  @Bean
  public EntityManager getEntitymanager()
  {
    return Persistence.createEntityManagerFactory("dev").createEntityManager();
  }
}
```

org.jsp.SpringHibernateApp.controller

LogoutController.java

```
package org.jsp.SpringHibernateApp.controller;

import javax.servlet.http.HttpSession;

import org.jsp.SpringHibernateApp.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class LogoutController
{
  @Autowired
  UserService service;

  @RequestMapping("/logout")
  public String logout(HttpSession session, Model model) {
    session.invalidate();
    model.addAttribute("message", "Logged Out");
    return "login";
  }

  @RequestMapping("/delete")
  public ModelAndView deleteuser(@RequestParam int id, ModelAndView view, HttpSession session) {
    service.DeleteUser(id);
    session.invalidate();
    view.setViewName("print");
    view.addObject("message", "user deleted");
    return view;
  }
}
```

UserController.java

```
package org.jsp.SpringHibernateApp.controller;

import org.jsp.SpringHibernateApp.dto.User;
import org.jsp.SpringHibernateApp.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.SessionAttributes;
import org.springframework.web.servlet.ModelAndView;

@Controller
@SessionAttributes(names = { "u" })
public class UserController {
  @Autowired
  private UserService service;

  @RequestMapping("/load")
  public ModelAndView load(ModelAndView view, @RequestParam int choice) {
    if (choice == 1)
      view.setViewName("regiser");
    if (choice == 2)
      view.setViewName("edit");
    view.addObject("user", new User());
    return view;
  }

  @RequestMapping("/reg")
  public ModelAndView SaveUser(@ModelAttribute User user, ModelAndView view) {
    user = service.SavaUser(user);
    view.setViewName("print");
    view.addObject("message", "User Saved with ID : " + user.getId());
    return view;
  }

  @RequestMapping(value = "/loginuser", method = RequestMethod.POST)
  public String login(@RequestParam long phone, @RequestParam String password, Model model) {
    User user = service.VerifyUser(phone, password);
    if (user != null) {
      model.addAttribute("u", user);
      return "home";
    } else {
      model.addAttribute("message", "Invalid Phone Number or Passwod");
      return "login";
    }
  }

  @RequestMapping("/edit")
  public ModelAndView updateUser(@ModelAttribute User user, ModelAndView view) {
    user = service.UpdateUser(user);
    view.addObject("user", user);
    view.setViewName("print");
    view.addObject("message", "user updated succesfully");
    return view;
  }
}
```

▼ persistence.xml

```
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
  http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
  version="2.1">

  <persistence-unit name="dev">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <shared-cache-mode>ENABLE_SELECTIVE</shared-cache-mode>      <!-- for caching -->
    <properties>
      <property name="javax.persistence.jdbc.driver"
        value="com.mysql.cj.jdbc.Driver" />
      <property name="javax.persistence.jdbc.url"
        value="jdbc:mysql://localhost:3306/SpringHibernate?createDatabaseIfNotExist=true" />
      <property name="javax.persistence.jdbc.user" value="root" />
      <property name="javax.persistence.jdbc.password"
        value="admin" />
      <property name="hibernate.show_sql" value="true" />

      <property name="hibernate.hbm2ddl.auto" value="update" />
      <property name="hibernate.dialect"
        value="org.hibernate.dialect.MySQL8Dialect" />
    </properties>
  </persistence-unit>
</persistence>
```

▼ web.xml

```
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>
  <welcome-file-list>
    <welcome-file>/WEB-INF/view/login.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>UserMvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>UserMvc</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>
```

UserMvc-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans.xsd
            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-context-3.0.xsd">
  <context:component-scan base-package="org.jsp"></context:component-scan>
  <context:annotation-config></context:annotation-config>

  <bean id="viewresolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/view/"></property>
    <property name="suffix" value=".jsp"></property>
  </bean>
</beans>
```

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<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">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.jsp.Spring-MVC</groupId>
  <artifactId>SpringHibernate-CRUD</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>SpringHibernate-CRUD Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet.jsp</groupId>
      <artifactId>jsp-api</artifactId>
      <version>2.2</version>
      <scope>provided</scope>
    </dependency>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.3.13</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.3.13</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.28</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.6.14.Final</version>
    </dependency>
  </dependencies>
  <build>
    <finalName>SpringHibernate-CRUD</finalName>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven
        defaults (may be moved to parent pom) -->
      <plugins>
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
        <!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin_bindings_for_war_packaging -->
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.22.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-war-plugin</artifactId>
          <version>3.2.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-install-plugin</artifactId>
          <version>2.5.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-deploy-plugin</artifactId>
          <version>2.8.2</version>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>
</project>
```

edit.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <c:if test="${u!=null }">
    <form:form action="edit" modelAttribute="user">
      <form:label path="id">ID</form:label>
      <form:input path="id" value="${u.getId() }" readonly="true" />
      <br>
      <form:label path="name">Name</form:label>
      <form:input path="name" value="${u.getName() }" />
      <br>
      <form:label path="age">Age</form:label>
      <form:input path="age" value="${u.getAge() }" />
      <br>
      <form:label path="phone">Phone</form:label>
      <form:input path="phone" value="${u.getPhone() }" />
      <br>
      <form:label path="password">Password</form:label>
      <form:input path="password" value="${u.getPassword() }" />
      <br>
      <form:button>Update</form:button>
    </form:form>
  </c:if>
</body>
</html>
```

home.jsp

```
<%@page import="org.jsp.SpringHibernateApp.dto.User"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <c:if test="${u!=null }">
    <h3>
      <a href="load?choice=2">Edit</a>
    </h3>
    <h3>
      <a href="view.jsp">Profile</a>
    </h3>
    <h3>
      <a href="delete?id=${u.getId() }">Delete</a>
    </h3>
    <h3>
      <a href="logout">Logout</a>
    </h3>
  </c:if>
</body>
</html>
```

login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
  <%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <c:if test="${message!=null }">
    <h2>${ message}</h2>
  </c:if>
  <form action="loginuser" method="post">
    <input type="tel" name="phone" placeholder="Enter Your Phone Number"><br>
    <input type="password" name="password" placeholder="Enter Your Password"><br>
    <input type="submit" value="LOGIN">
  </form>
  <h3><a href="load?choice=1">Click Here To Register</a></h3>
</body>
</html>
```

print.jsp

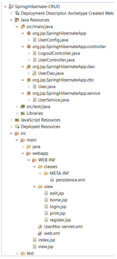
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
${message }
</body>
</html>
```

register.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <form:form modelAttribute="user" action="reg" method="post">
    <form:label path="name">Name</form:label>
    <form:input path="name" />
    <br>
    <form:label path="age">Age</form:label>
    <form:input path="age" />
    <br>
    <form:label path="phone">Phone</form:label>
    <form:input path="phone" />
    <br>
    <form:label path="password">Password</form:label>
    <form:input path="password" />
    <br>
    <form:button>Register</form:button>
  </form:form>
</body>
</html>
```

view.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
  pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <c:if test="${u!=null }">
    <h1>name:${u.getName() }</h1>
  </c:if>
</body>
</html>
```



```
SpringHibernate-CRUD
  Deployment Descriptor: Archetype Created Web
  Java Resources
    src/main/java
      org.jsp.SpringHibernateApp
        UserConfig.java
      org.jsp.SpringHibernateApp.controller
        LogoutController.java
        UserController.java
      org.jsp.SpringHibernateApp.dao
        UserDao.java
      org.jsp.SpringHibernateApp.dto
        User.java
      org.jsp.SpringHibernateApp.service
        UserService.java
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
    main
      java
      webapp
        WEB-INF
          classes
            META-INF
              persistence.xml
          view
            edit.jsp
            home.jsp
            login.jsp
            print.jsp
            register.jsp
          UserMvc-servlet.xml
          web.xml
        index.jsp
        view.jsp
    test
```