SPRING MVC FORM HANDLING EXAMPLE

 Copyright © tutorialspoint.com

The following example show how to write a simple web based application which makes use of HTML forms using Spring Web MVC framework. To start with it, let us have working Eclipse IDE in place and follow the following steps to develope a Dynamic Form based Web Application using Spring Web Framework:

Step	Description
1	Create a *Dynamic Web Project* with a name *HelloWeb* and create a package *com.tutorialspoint* under the *src* folder in the created project.
2	Drag and drop below mentioned Spring and other libraries into the folder *WebContent/WEB-INF/lib*.
3	Create a Java classes *Student* and *StudentController* under the *com.tutorialspoint* package.
4	Create Spring configuration files *Web.xml* and *HelloWeb-servlet.xml* under the *WebContent/WEB-INF* folder.
5	Create a sub-folder with a name *jsp* under the *WebContent/WEB-INF* folder. Create a view files *student.jsp* and *result.jsp* under this sub-folder.
6	The final step is to create the content of all the source and configuration files and export the application as explained below.

Here is the content of **Student.java** file:

```
package com.tutorialspoint;

public class Student {
   private Integer age;
   private String name;
   private Integer id;

   public void setAge(Integer age) {
      this.age = age;
   }
   public Integer getAge() {
      return age;
   }

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

   public void setId(Integer id) {
      this.id = id;
   }
   public Integer getId() {
      return id;
   }
}
```

Following is the content of **StudentController.java** file:

```
package com.tutorialspoint;
```

```
import org.springframework.stereotype.Controller;
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.servlet.ModelAndView;
import org.springframework.ui.ModelMap;

@Controller
public class StudentController {

   @RequestMapping(value = "/student", method = RequestMethod.GET)
   public ModelAndView student() {
      return new ModelAndView("student", "command", new Student());
   }

   @RequestMapping(value = "/addStudent", method = RequestMethod.POST)
   public String addStudent(@ModelAttribute("SpringWeb")Student student,
   ModelMap model) {
      model.addAttribute("name", student.getName());
      model.addAttribute("age", student.getAge());
      model.addAttribute("id", student.getId());

      return "result";
   }
}
```

Here the first service method **student**, we have passed a blank **Student** object in the ModelAndView object with name "command" because the spring framework expects an object with name "command" if you are using <form:form> tags in your JSP file. So when **student** method is called it returns **student.jsp** view.

Second service method **addStudent** will be called against a POST method on the **HelloWeb/addStudent** URL. You will prepare your model object based on the submitted information. Finally a "result" view will be returned from the service method, which will result in rendering result.jsp

Following is the content of Spring Web configuration file **web.xml**

```
<web-app
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

    <display-name>Spring MVC Form Handling</display-name>

    <servlet>
        <servlet-name>HelloWeb</servlet-name>
        <servlet-class>
           org.springframework.web.servlet.DispatcherServlet
        </servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>HelloWeb</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

</web-app>
```

Following is the content of another Spring Web configuration file **HelloWeb-servlet.xml**

```
<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.tutorialspoint" />

   <bean >
      <property name="prefix" value="/WEB-INF/jsp/" />
      <property name="suffix" value=".jsp" />
   </bean>

</beans>
```

Following is the content of Spring view file **student.jsp**

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
    <title>Spring MVC Form Handling</title>
</head>
<body>

<h2>Student Information</h2>
<form:form method="POST" action="/HelloWeb/addStudent">
   <table>
    <tr>
        <td><form:label path="name">Name</form:label></td>
        <td><form:input path="name" /></td>
    </tr>
    <tr>
        <td><form:label path="age">Age</form:label></td>
        <td><form:input path="age" /></td>
    </tr>
    <tr>
        <td><form:label path="id">id</form:label></td>
        <td><form:input path="id" /></td>
    </tr>
    <tr>
        <td colspan="2">
            <input type="submit" value="Submit"/>
        </td>
    </tr>
</table>
</form:form>
</body>
</html>
```

Following is the content of Spring view file **result.jsp**

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
    <title>Spring MVC Form Handling</title>
</head>
<body>

<h2>Submitted Student Information</h2>
   <table>
    <tr>
        <td>Name</td>
        <td>${name}</td>
    </tr>
    <tr>
        <td>Age</td>
        <td>${age}</td>
    </tr>
    <tr>
```

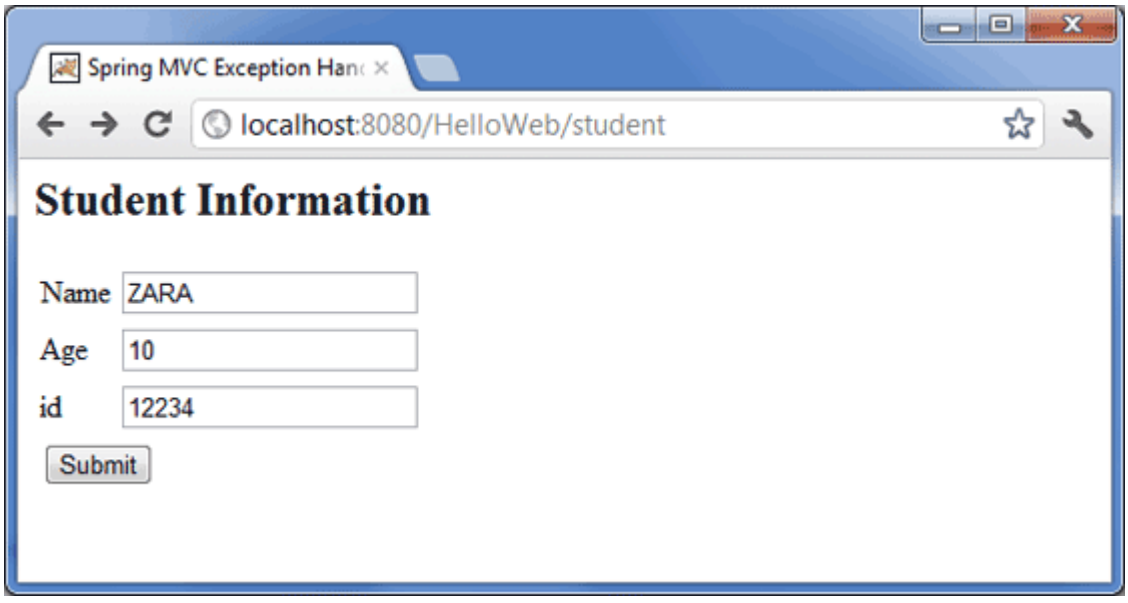
```
        <td>ID</td>
        <td>${id}</td>
    </tr>
</table>
</body>
</html>
```

Finally, following is the list of Spring and other libraries to be included in your web application. You simply drag these files and drop them in **WebContent/WEB-INF/lib** folder.

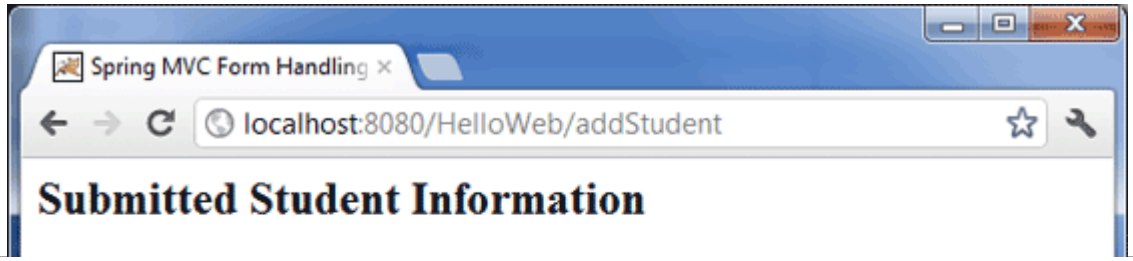
- commons-logging-x.y.z.jar
- org.springframework.asm-x.y.z.jar
- org.springframework.beans-x.y.z.jar
- org.springframework.context-x.y.z.jar
- org.springframework.core-x.y.z.jar
- org.springframework.expression-x.y.z.jar
- org.springframework.web.servlet-x.y.z.jar
- org.springframework.web-x.y.z.jar
- spring-web.jar

Once you are done with creating source and configuration files, export your application. Right click on your application and use **Export > WAR File** option and save your **SpringWeb.war** file in Tomcat's *webapps* folder.

Now start your Tomcat server and make sure you are able to access other web pages from webapps folder using a standard browser. Now try a URL **http://localhost:8080/SpringWeb/student** and you should see the following result if everything is fine with your Spring Web Application:



After submitting required information click on submit button to submit the form. You should see the following result if everything is fine with your Spring Web Application:



Name ZARA
Age 10
ID 12234

Loading [MathJax]/jax/output/HTML-CSS/jax.js