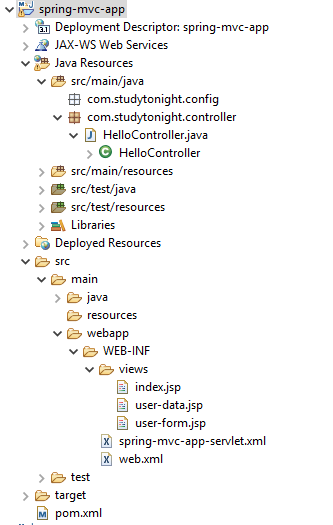
Spring MVC Handling Form Data

In this article, we are going to create a Spring application and fetching HTML form data into a JSP page.

We will use **param** implicit object to fetch form data. Below is our project structure that shows the available files in our project.

Project Structure



**// HelloController.java**

This is a controller class that contains two methods showForm() and showdata(). The showForm() method is mapped with userForm request and used to return a user-form.

package com.studytonight.controller;

import org.springframework.stereotype.Controller;

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

import org.springframework.web.bind.annotation.RequestMethod;

@Controller

public class HelloController {

@RequestMapping(value = "userform", method = RequestMethod.GET)

public String showForm() {

return "user-form";

}

@RequestMapping(value = "showdata", method = RequestMethod.GET)

public String showdata() {

return "user-data";

}

}

**// user-form.jsp**

This is our HTML form page that takes user input and submits it to the server at showdata mapping. This request will be handled by the controller(HomeController).

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>User Form</title>

</head>

<body>

<form action="showdata" method="Get">

<input type="text" name="user\_name"><br/><br/>

<input type="submit" />

</form>

</body>

</html>

**// user-data.jsp**

This page shows the input entered by the user in HTML form. The controller returns this page as a response to the client. This page uses the **Expression Language** to fetch the user form data. Here, **param** is an implicit object of JSP that contains all the request data.

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Hello JSP</title>

</head>

<body>

<strong>Hello,</strong>

${param.user\_name}

</body>

</html>

These are Spring configuration XML files that configure views files and dispatcher servlet to provide a front controller for the application.

**// spring-mvc-app-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"

xmlns:mvc="http://www.springframework.org/schema/mvc"

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.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc.xsd">

<!-- Step 3: Add support for component scanning -->

<context:component-scan base-package="com.studytonight.controller" />

<!-- Step 4: Add support for conversion, formatting and validation support -->

<mvc:annotation-driven/>

<!-- Step 5: Define Spring MVC view resolver -->

<bean

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

<property name="prefix" value="/WEB-INF/views/" />

<property name="suffix" value=".jsp" />

</bean>

</beans>

**// web.xml**

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

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

id="WebApp\_ID" version="3.1">

<display-name>spring-mvc-app</display-name>

<absolute-ordering />

<!-- Spring MVC Configs -->

<!-- Step 1: Configure Spring MVC Dispatcher Servlet -->

<servlet>

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

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

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

<param-value>/WEB-INF/spring-mvc-app-servlet.xml</param-value>

</init-param>

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

</servlet>

<!-- Step 2: Set up URL mapping for Spring MVC Dispatcher Servlet -->

<servlet-mapping>

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

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

</servlet-mapping>

</web-app>

Copy

**// pom.xml**

This file contains all the dependencies of this project such as spring jars, servlet jars, etc. Put these dependencies into your project to run the application.

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.studytonight</groupId>

<artifactId>spring-mvc-app</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.10.0.pr3</version>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.xml.bind/jaxb-api -->

<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.0</version>

</dependency>

</dependencies>

<properties>

<spring.version>5.2.8.RELEASE</spring.version>

</properties>

<build>

<plugins>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.3</version>

<configuration>

<warSourceDirectory>WebContent</warSourceDirectory>

</configuration>

</plugin>

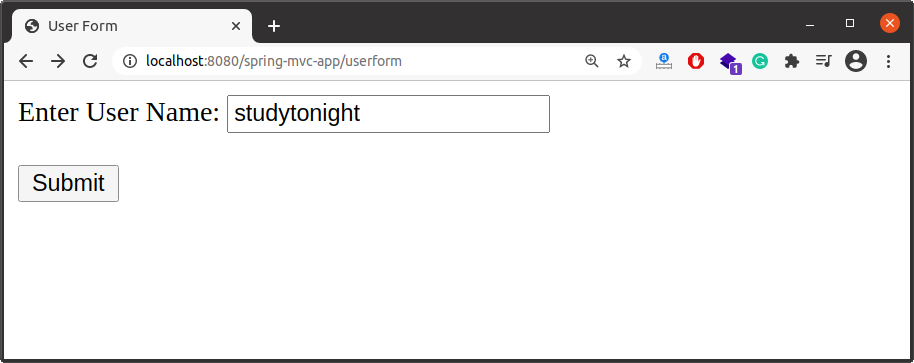
</plugins>

</build>

</project>

Run the Application

After successfully completing the project and adding the dependencies run the application and you will get the output as below. Submit the form after entering the user name.



This is the second JSP page that shows the user name submitted by the above user name.

