**ITL-3 Practical 2**

**Q:** Write a program to redirect a request using a dynamic approach

**A:**

The following example show how to write a simple web-based application which makes use of redirect to transfer a http request to another page. To start with, let us have a working Eclipse IDE in place and take the following steps to develop 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 class *WebController* 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 view files *index.jsp* and *final.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 **WebController.java** file

package com.tutorialspoint;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class WebController {

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

public String index() {

return "index";

}

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

public String redirect() {

return "redirect:finalPage";

}

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

public String finalPage() {

return "final";

}

}

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

<web-app id = "WebApp\_ID" version = "2.4"

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 Page Redirection</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**

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

<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 id = "viewResolver"

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

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

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

</bean>

</beans>

Following is the content of Spring view file **index.jsp**. This will be a landing page, this page will send a request to access redirect service method which will redirect this request to another service method and finally a **final.jsp** page will be displauyed.

<%@taglib uri = "http://www.springframework.org/tags/form" prefix = "form"%>

<html>

<head>

<title>Spring Page Redirection</title>

</head>

<body>

<h2>Spring Page Redirection</h2>

<p>Click below button to redirect the result to new page</p>

<form:form method = "GET" action = "/HelloWeb/redirect">

<table>

<tr>

<td>

<input type = "submit" value = "Redirect Page"/>

</td>

</tr>

</table>

</form:form>

</body>

</html>

Following is the content of Spring view file **final.jsp**. This is the final redirected page.

<%@taglib uri = "http://www.springframework.org/tags/form" prefix = "form"%>

<html>

<head>

<title>Spring Page Redirection</title>

</head>

<body>

<h2>Redirected Page</h2>

</body>

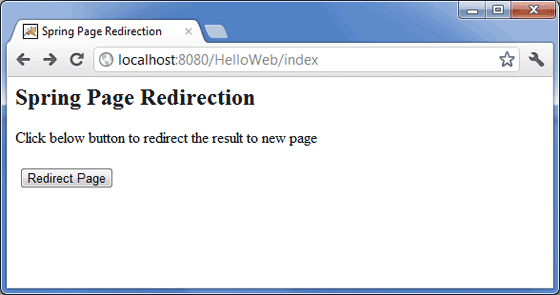
</html>

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 HelloWeb.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. Try a URL http://localhost:8080/HelloWeb/index and you should see the following result if everything is fine with your Spring Web Application.



Click the "Redirect Page" button to submit the form and to get the final redirected page. You should see the following result if everything is fine with your Spring Web Application.

