- Java
  - JAXB Tutorial
    - What is JAXB
    - JAXB Marshalling Example
    - JAXB UnMarshalling Example
  - Collections
    - Java ArrayList Class
    - Java Vector Class
    - Java HashSet Class
- Spring MVC Tutorial
  - Quick Start
    - Flow Diagram
    - Hello World Example
    - Form Handling Example
  - Handler Mapping
    - BeanNameUrlHandlerMapping
    - ControllerClassNameHandlerMapping
    - SimpleUrlHandlerMapping
  - Validation & Exception Handling
    - Validation+Annotations
    - Validation+ResourceBundle
    - @ExceptionHandler
    - @ControllerAdvice
    - Custom Exception Handling
  - Form Tag Library
    - Textbox Example
    - TextArea Example
    - Password Example
    - Dropdown Box Example
    - Checkboxes Example
    - Radiobuttons Example
    - HiddenValue Example
  - Misc
    - Change Config file name
- REST Tutorial
  - JAX-RS REST @PathParam Example
  - JAX-RS REST @QueryParam Example
  - JAX-RS REST @DefaultValue Example
  - JAX-RS REST @Context Example
  - JAX-RS REST @MatrixParam Example
  - JAX-RS REST @FormParam Example
  - JAX-RS REST @Produces Example
  - JAX-RS REST @Consumes Example
  - JAX-RS REST @Produces both XML and JSON Example
  - JAX-RS REST @Consumes both XML and JSON Example
- Miscellaneous
  - JSON Parser
    - Read a JSON file
    - Write JSON object to File
    - Read / Write JSON using GSON
    - Java Object to JSON using JAXB
  - CSV Parser
    - Read / Write CSV file
    - Read/Parse/Write CSV File – OpenCSV
    - Export data into a CSV File
    - CsvToBean and BeanToCsv – OpenCSV
- Search the site ... Search

JavaInterviewPoint

Java Development Tutorials

Spring MVC SimpleFormController Example

October 5, 2015 by javainterviewpoint Leave a Comment

We have already learnt about **Form handling in Spring** using **@Controller annotation**. In this article we will learn how to use **SimpleFormController**(Deprecated in Spring 3) instead of **@Controller** to handle a **Spring MVC Form**.

When we take a look into the configuration file, the mapping for the controller will be made like below

```
<bean name="/ShowRegistrationForm.htm" class="com.javainterviewpoint.SimpleFormControllerExample"
 p:formView="StudentRegistrationForm"/>
```

Here we can see we have mentioned the Spring container to use **SimpleFormControllerExample** as the controller for request **"ShowRegistrationForm.htm"**

Folder Structure :

1. Create a *Dynamic Web Project* **"SpringMVCFormHandling"** and create a package for our src files ***"com.javainterviewpoint"***
2. Place the Spring 3 jar files under *WEB-INF/Lib*

   commons-logging-1.1.1.jar
   log4j-1.2.16.jar
   slf4j-api-1.7.5.jar
   slf4j-log4j12-1.7.5.jar
   spring-aspects-3.2.4.RELEASE.jar
   spring-beans-3.2.4.RELEASE.jar
   spring-context-3.2.4.RELEASE.jar
   spring-core-3.2.4.RELEASE.jar
   spring-expression-3.2.4.RELEASE.jar
   spring-web-3.2.4.RELEASE.jar
   spring-webmvc-3.2.4.RELEASE.jar

3. Create the Java classes ***SimpleFormControllerExample.java*** and ***Student.java*** under *com.javainterviewpoint* folder.
4. Place the ***SpringConfig-servlet.xml*** *and* ***web.xml*** under the *WEB-INF directory*
5. View files ***StudentRegistrationForm.jsp*** and ***studentSuccess.jsp*** are put under the sub directory under ***WEB-INF/Jsp***
6. Place the ***redirector.jsp*** under the **WebContent** folder which will act as our index page.

SimpleFormControllerExample.java

- Our **SimpleFormControllerExample** class extends **SimpleFormController** class and overrides **"onSubmit()"** method.
- In the constructor of our class**(SimpleFormControllerExample)** we will be setting **CommandClass** and **CommandName** as our **Student** Class.
- Inside the **onSubmit()** method, We will get the command object and will type cast to **Student Type** and will create a **ModelAndView** object which has the redirecting page**(studentSuccess)** and Command **"st"** along with it.

```
package com.javainterviewpoint;

import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.SimpleFormController;

@SuppressWarnings("deprecation")
public class SimpleFormControllerExample extends SimpleFormController
{
    public SimpleFormControllerExample()
    {
        setCommandClass(Student.class);
        setCommandName("st");
    }
    @Override
    protected ModelAndView onSubmit(Object command) throws Exception {
        Student st = (Student)command;
        return new ModelAndView("studentSuccess","st",st);
    }
}
```

Student.java

Student class act as our Model class here. It has two properties **studentId** & **studentName** and their corresponding **getters** and **setters**.

```
package com.javainterviewpoint;

public class Student
{
    private int studentId;
    private String studentName;
    public int getStudentId() {
        return studentId;
    }
    public void setStudentId(int studentId) {
        this.studentId = studentId;
    }
    public String getStudentName() {
        return studentName;
    }
    public void setStudentName(String studentName) {
        this.studentName = studentName;
    }

}
```

redirector.jsp

redirector.jsp will be our index page, which redirects to **"ShowRegistrationForm.htm"** whose mapping will be made in **SpringConfig-servlet.xml**

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

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
  </head>
  <body>
    <%
       response.sendRedirect("ShowRegistrationForm.htm");
    %>
  </body>
</html>
```

StudentRegistrationForm.jsp

StudentRegistrationForm will have the form where user enters the student details

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
 pageEncoding="ISO-8859-1"%>
 <%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

```
        <title>Insert title here</title>
      </head>
      <body>
        <form:form method = "POST" commandName="st">
          Student Id : <form:input path="studentId"/><br>
          Student Name: <form:input path="studentName"/><br>
          <input type="submit">
        </form:form>
      </body>
    </html>
```

studentSuccess.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
 pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
        <title>Insert title here</title>
    </head>
    <body>
        <h2> Welcome to ${msg}</h2>
    </body>
</html>
```

web.xml

The web.xml has everything about the application that a server needs to know, which is placed under the WEB-INF directory. *<servlet-name>* contains the name of the SpringConfiguration , when the *DispatcherServlet* is initialized the framework will try to load a configuration file *"[servlet-name]-servlet.xml"* under the WEB-INF directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee%20http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
   <display-name>SpringMVCFormHandling</display-name>
   <welcome-file-list>
       <welcome-file>index.html</welcome-file>
       <welcome-file>index.htm</welcome-file>
       <welcome-file>index.jsp</welcome-file>
       <welcome-file>default.html</welcome-file>
       <welcome-file>default.htm</welcome-file>
       <welcome-file>default.jsp</welcome-file>
   </welcome-file-list>
   <servlet>
       <servlet-name>SpringConfig</servlet-name>
       <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
   </servlet>
   <servlet-mapping>
      <servlet-name>SpringConfig</servlet-name>
      <url-pattern>/</url-pattern>
   </servlet-mapping>
</web-app>
```

SpringConfig-servlet.xml

- The **SpringConfig-servlet.xml** is also placed under the WEB-INF directory.
- Here we have configured **SimpleFormContollerExample** as the controller for the request **"/ShowRegistrationForm.htm"**
- We have also mapped the view page which the request has to be redirected to **"StudentRegistrationForm"** using **"p:formView"** attribute.

```
<?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:p="http://www.springframework.org/schema/p"
xsi:schemaLocation="http://www.springframework.org/schema/beans%20http://www.springframework.org/schema/beans/spring-beans.xsd">

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

     <bean name="/ShowRegistrationForm.htm" class="com.javainterviewpoint.Simple       FormControllerExample" p:formView="StudentRegistrationForm"/>
</beans>
```

Output

StudentRegistrationForm

Success Page

Related posts:

1. Spring MVC HiddenValue Example
2. Spring MVC Multiple submit buttons in a single form
3. Spring 4 – Spring MVC Hello World Example
4. Spring MVC BeanNameUrlHandlerMapping Example

Filed Under: J2EE, Java, Spring, Spring MVC Tagged With: Controller, MVC, SimpleFormController, SimpleFormController Example, Spring, Spring MVC

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

Java Interview Point
Like Page 754 likes

Core Java Tutorial

Java Basics

- **JVM Architecture**
- **Object in Java**
- **Class in Java**
- **How to Set Classpath for Java in Windows**
- **Components of JDK**
- **Decompiling a class file**
- **Use of Class.forName in java**
- **Use Class.forName in SQL JDBC**

Oops Concepts

- Inheritance in Java
- Types of Inheritance in Java
- Single Inheritance in Java
- Multiple Inheritance in Java
- Multilevel Inheritance in Java
- Hierarchical Inheritance in Java
- Hybrid Inheritance in Java

- Polymorphism in Java – Method Overloading and Overriding
- Types of Polymorphism in java
- Method Overriding in Java
- Can we Overload static methods in Java
- Can we Override static methods in Java
- Java Constructor Overloading
- Java Method Overloading Example

- Encapsulation in Java with Example

- Constructor in Java
- Constructor in an Interface?
- Parameterized Constructor in Java
- Constructor Chaining with example
- What is the use of a Private Constructors in Java

- Interface in Java
- What is Marker Interface
- Abstract Class in Java

Java Keywords

- Java this keyword
- Java super keyword
- Final Keyword in Java
- static Keyword in Java
- Static Import
- Transient Keyword

Miscellaneous

- newInstance() method
- How does Hashmap works internally in Java

- Java Ternary operator
- How System.out.println() really work?
- Autoboxing and Unboxing Examples
- Serialization and Deserialization in Java with Example
- Generate SerialVersionUID in Java
- How to make a class Immutable in Java
- Differences betwen HashMap and Hashtable
- Difference between Enumeration and Iterator ?
- Difference between fail-fast and fail-safe Iterator
- Difference Between Interface and Abstract Class in Java
- Difference between equals() and ==
- Sort Objects in a ArrayList using Java Comparable Interface
- Sort Objects in a ArrayList using Java Comparator

Useful Links

- Spring 4.1.x Documentation
- Spring 3.2.x Documentation
- Spring 2.5.x Documentation
- Java 6 API
- Java 7 API
- Java 8 API
- Java EE 5 Tutorial
- Java EE 6 Tutorial
- Java EE 7 Tutorial
- Maven Repository
- Hibernate ORM

javainterviewpoint.com is a tech blog dedicated to all Java/J2EE developers and Web Developers. We publish useful tutorials on Java, J2EE and all latest frameworks.All examples and tutorials posted here are very well tested in our development environment.Connect with us on **facebook**Privacy Policy | Contact Us