My Tech Blog

Java Web Service SOA

- Home
- **My Learnings**
- **Java**
- **XStream**
- **ESB**
- **BPEL**
- **Webservices**
- **Hibernate**
- **SQL**
- **J2EE**
-

Type text to search here...

Home > Servlet & JSP > **Simple Webapplication using Servlet and JSP**

Simple Webapplication using Servlet and JSP

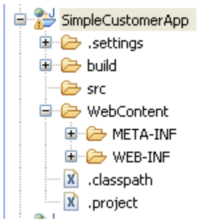
October 13, 2011 sabapathy Leave a comment Go to comments

Here in this post we will see simple web application using Servlet and JSP where we are going to give customer details in JSP page then get the customer details in servlet and show the customer details inserted in Welcome page.

Environment used:
1. Eclipse 3.3
2. JDK6
3. Tomcat Servler 6.x

1. Create "Dynamic Web Project" with **Dynamic web module version – 2.5** Then Click **Finish**.

Web application structure in eclipse will look like this.



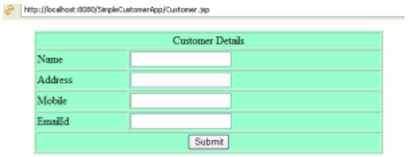
```
SimpleCustomerApp
    .settings
    build
    src
    WebContent
        META-INF
        WEB-INF
    .classpath
    .project
```

WebAppStructure

2. Create a **Customer.jsp** page inside **WebContent** folder of web application structure.

```
<%@ 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>Customer Details</title>
</head>
<body>
<form action="insertCustomer" method="post">
<table align="center" bgcolor="#99FFCC" border="1" width="70%">
        <tr>
                <td colspan="2" align="center">Customer Details </td>
        </tr>
        <tr>
                <td>Name </td>
                <td><input type="text" name="name" maxlength="25"></td>
        </tr>
        <tr>
                <td>Address </td>
                <td><input type="text" name="address" maxlength="40"></td>
        </tr>
        <tr>
                <td>Mobile </td>
                <td><input type="text" name="mobile" maxlength="10"></td>
        </tr>
        <tr>
                <td>EmailId </td>
                <td><input type="text" name="emailid" maxlength="30"></td>
        </tr>
        <tr>
                <td colspan="2" align="center"><input type="submit" value="Submit"></td>
        </tr>
</table>
</form>
</body>
</html>
```

This customer page will look like the below.



CustomerPage

3. Create a **InsertCustomerServlet.java** servlet class. While submitting the form from the "Customer.jsp" page the servlet will get the form value in **HttpServletRequest** object .

```
package com.room.sample.servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.room.sample.view.Customer;

public class InsertCustomerServlet extends HttpServlet{

        private static final long serialVersionUID = 1L;
        public void doPost(HttpServletRequest request, HttpServletResponse response){
                System.out.println("----- InsertCustomerServlet -----");
                try {
                // Get the customer value submitted from Customer.jsp page through HttpServletRequest object
                        String name=request.getParameter("name");
                        String address=request.getParameter("address");
                        String mobile=request.getParameter("mobile");
                        String emailid=request.getParameter("emailid");

                        //Set the Customer values into Customer Bean or POJO(Plain Old Java Object) class
                        Customer customer=new Customer();
                        customer.setName(name);
                        customer.setAddress(address);
                        customer.setMobile(Long.valueOf(mobile));
                        customer.setEmailid(emailid);

                        RequestDispatcher dispatcher=request.getRequestDispatcher("/Welcome.jsp");
                        //Set the customer instance into request.Then only the customer object
                        //will be available in the Welcome.jsp page
                        request.setAttribute("cust",customer);
                        dispatcher.forward(request, response);
                } catch (ServletException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                }catch(Exception e){
                        e.printStackTrace();
                }

        }

}
```

4. Bean Class to store information received in Servlet class.

```
package com.room.sample.view;

import java.io.Serializable;

public class Customer implements Serializable{

        private static final long serialVersionUID = 1L;
        private String name;
        private String address;
        private Long mobile;
        private String emailid;
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public String getAddress() {
                return address;
        }
        public void setAddress(String address) {
                this.address = address;
        }
        public Long getMobile() {
                return mobile;
        }
        public void setMobile(Long mobile) {
                this.mobile = mobile;
        }
        public String getEmailid() {
                return emailid;
        }
        public void setEmailid(String emailid) {
                this.emailid = emailid;
        }

}
```

5. Create **Welcome.jsp** page to display the customer details processed.

```
<%@page import="com.room.sample.view.Customer"%>
<%@ 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>View Customer Details</title>
</head>
<body>
<%
        Customer customer=(Customer)request.getAttribute("cust");
%>
<table align="center" bgcolor="#FFFFCC" border="1" width="70%">
        <tr>
                <td colspan="2" align="center"><%="Welcome "+customer.getName()+" !!!!. Your details Processed." %></td>
        </tr>
        <tr>
                <td>Name </td>
                <td><%=customer.getName()%></td>
        </tr>
        <tr>
                <td>Address </td>
                <td><%=customer.getAddress() %></td>
        </tr>
        <tr>
                <td>Mobile </td>
                <td><%=String.valueOf(customer.getMobile()) %></td>
        </tr>
        <tr>
                <td>EmailId </td>
                <td><%=customer.getEmailid() %></td>
        </tr>
</table>
</body>
</html>
```

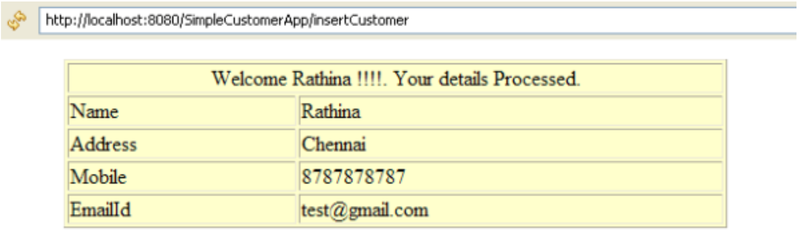
6. Connect the flow using **web.xml**. For a servlet defined , the element value under element and element value under should be same.
Then only the requested action **“insertCustomer”** can find the corresponsing servlet to be executed from **web.xml** file.

```
<?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-
  <display-name>Simple Customer Application</display-name>
  <welcome-file-list>
    <welcome-file>Customer.jsp</welcome-file>
  </welcome-file-list>

  <servlet>
        <servlet-name>customer</servlet-name>
        <servlet-class>com.room.sample.servlet.InsertCustomerServlet</servlet-class>
  </servlet>

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

Welcome.jsp page will look like the below.



http://localhost:8080/SimpleCustomerApp/insertCustomer

Welcome Rathina !!!!. Your details Processed.	
Name	Rathina
Address	Chennai
Mobile	8787878787
EmailId	test@gmail.com

Welcome

★ Like

One blogger likes this.

Related

OSB11G calling RESTful Web Service Example In "OSB"	RESTful web service sample in Tomcat In "Restful"	Difference between ServletConfig and ServletContext In "Servlet & JSP"

Categories: Servlet & JSP Tags: building simple webapplication, building web application, building web application using servlet and jsp, jsp, servlet, servlet and JSP, simple web application, web ap;lication, web application using servlet and jsp
Comments (4) Trackbacks (0) Leave a comment Trackback

1. Ksenia
February 4, 2013 at 5:10 pm
Reply

   Thank you!!!:)

2. Bhanu
February 25, 2013 at 12:44 pm
Reply

   good example

3. Farag Elfadaly
March 21, 2013 at 2:35 am
Reply

   example explain jsp and servlet simply
thank yoy

4. panic anxiety disorder
May 17, 2013 at 3:20 am
Reply

   This piece of writing is truly a pleasant one it helps new internet visitors, who are wishing in favor of blogging.

1. No trackbacks yet.

Leave a Reply

Enter your comment here...

[Difference between Statement and PreparedStatement in JDBC with examples](#)
[Connect to Server using WLST](#)
RSS feed

Categories

- ANT (7)
- Eclipse (1)
    - Decompiler (1)
- General (1)
- Hibernate (5)
- J2EE (5)
    - EJB3 (1)
    - JDBC (1)
    - Servlet & JSP (3)
- Java (8)
    - sql (2)
    - XStream (1)
- JAXB (1)
- Oracle (2)
- Restful (1)
- SOA (36)
    - Adapter (1)
    - BAM (4)
    - BPEL (12)
    - DVM (2)
    - Exceptions_and_Solutions (3)
    - Installation (2)
        - RCU (1)
    - JDeveloper (1)
    - MDS (3)
    - Mediator (1)
    - OSB (5)
    - WLST (1)
- Weblogic (1)
- webservice (5)
- XSLT (5)

Stats

My Calendar

October 2011

M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

« Sep Nov »

Archieves

Select Month

Meta

- Register
- Log in
- Entries RSS
- Comments RSS
- Create a free website or blog at WordPress.com.

Recent Posts

- **OSB11G Dynamic Routing Sample**
- **OSB11G calling RESTful Web Service Example**
- **OSB11G Java Callout Example**
- **OSB11G – Sample With Assign, Replace, Insert, ServiceCallOut, Log and XQuery**
- **OSB11G_SayHello Sample (WSDL Based)**

Pages

- **BPEL**
- **ESB**
- **Hibernate**
- **J2EE**
- **Java**
- **My Learnings**
- **SQL**
- **Webservices**
- **XStream**

Top

Create a free website or blog at WordPress.com. The INove Theme.