**Angular JS – Day 5**

**(Communication with Server Programs)**

**AJAX, JSON, Servlet, JDBC, RESTful web service**

**Exercises**

1. **Write a simple Java servlets with doGet() and doPost() methods and call these methods from Angular JS and pass the data got from servlet into an angular directive.**

**What you will learn by doing this exercise:**

Angular – servlet communication,

Application of Angular filter mechanism

Formatting the data.

Step 1: Write a simple HelloWorld servlet in Eclipse (by creating dynamic web project)

Unit test both the methods.

package com.oracle;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class HelloServlet

\*/

public class HelloServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public HelloServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter pw=response.getWriter();

pw.println("Hello World Servlet in GET method");

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter pw=response.getWriter();

pw.println("Hi all, this is a POST method in HelloServlet");

}

}

Step 2:

Create MyForm.html with angular script as follows:

<html lang=*"en"*>

<head>

<title>Angular Servlet communication</title>

<script src=*"angular.min.js"*></script>

</head>

<body ng-app=*"myApp"* ng-controller=*"GuruController"*>

<h1> Communication with server </h1>

<ol>

<li>

{{myData | lowercase}}

</li>

</ol>

</body>

<script>

**var** app = angular.module('myApp', []);

app.controller('GuruController', **function**($scope,$http)

{

var response = $http.post("http://localhost:8081/HelloWorldServlet/Hello");

response.success(**function**(data, status, headers, config) {

$scope.myData = data;

});

response.error(**function**(data, status, headers, config) {

alert('failure');

});

});

</script>

</html>

Step 3:

Right click MyForm.html and run on the web server.

You can see that the data returned from the servlet will be displayed in uppercase (Data Formatting).

Also, the model is set in <ol> view.

1. **Write a simple Restful web service that returns some data “Hello UST”. Produce the data in different formats (plain text, xml format, html format, JSON format). Call this from a simple HTML file.**

**What you will learn by doing this exercise:**

Writing simple Restful web service program

Producing different types of data

Calling the web service from HTML page.

**Requirement:**

You need a jersey framework APIs for running this program

Download and keep the .jar files in WEB-INF/lib folder of the web application

**web.xml:**

<?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 http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* id=*"WebApp\_ID"* version=*"2.5"*>

<display-name>GuruWS</display-name>

<servlet>

<servlet-name>Jersey REST Service</servlet-name>

<servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>

<init-param>

<param-name>jersey.config.server.provider.packages</param-name>

<param-value>com.ust\_global</param-value>

</init-param>

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

</servlet>

<servlet-mapping>

<servlet-name>Jersey REST Service</servlet-name>

<url-pattern>/restGuru/\*</url-pattern>

</servlet-mapping>

</web-app>

**2. Create Simple java class (this is data class)**

**package** com.ust\_global;

**import** javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement

**public** **class** MyWork {

**private** String summary;

**private** String description;

**public** String getSummary() {

**return** summary;

}

**public** **void** setSummary(String summary) {

**this**.summary = summary;

}

**public** String getDescription() {

**return** description;

}

**public** **void** setDescription(String description) {

**this**.description = description;

}

}

**2. Create Service class GetResource.java:**

package com.ust\_global;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

@Path("/MyWorks")

public class GetResource {

// This method is called if XMLis request

@GET

@Produces({MediaType.APPLICATION\_JSON})

public MyWork getJSON() {

MyWork mw=new MyWork();

mw.setSummary("This is my first mywork JSON");

mw.setDescription("This is my first mywork JSON");

return mw;

}

// // This method is called if XMLis request

// @GET

// @Produces({ MediaType.APPLICATION\_XML})

// public MyWork getXML() {

// MyWork mw=new MyWork();

// mw.setSummary("This is my first mywork XML");

// mw.setDescription("This is my first mywork XML");

// return mw;

// }

// This can be used to test the integration with the browser

@GET

@Produces({ MediaType.TEXT\_XML })

public MyWork getHTML() {

MyWork mw=new MyWork();

mw.setSummary("This is my first mywork");

mw.setDescription("This is my first mywork");

return mw;

}

}

Give the following URL to get the data in JSON format:

<http://localhost:8081/RestFulAutomatic/restGuru/MyWorks>

1. **Angular Restful web service communication**

**What you will learn by doing this exercise:**

Calling the RESTful web service from AngularJS program.

Create MyForm.html with angular script as follows.

Call this web service from this html page.

<html lang=*"en"*>

<head>

<title>Angular Servlet communication</title>

<script src=*"angular.min.js"*></script>

</head>

<body ng-app=*"myApp"* ng-controller=*"GuruController"*>

<h1> Communication with web webservice </h1>

<ol>

<li>

{{myData | lowercase}}

</li>

</ol>

</body>

<script>

**var** app = angular.module('myApp', []);

app.controller('GuruController', **function**($scope,$http)

{

var response = $http.post("<http://localhost:8081/RestFulAutomatic/restGuru/MyWorks>

");

response.success(**function**(data, status, headers, config) {

$scope.myData = data;

});

response.error(**function**(data, status, headers, config) {

alert('failure');

});

});

</script>

</html>

1. **Develop an application that makes the following communication:**

**JavaScript – AJAX – Servlet – JDBC - JSON**

**What you will learn by doing this exercise:**

How to make AJAX call to server side program

How to make JDBC (database) call to Oracle database

How to convert the data into JSON object

How to convert JSON object into JavaScript object

**Step 1: Form.html**

<html>

<div id=*"test"* style="color:*white*;background-color:*blue*; text-align:*left*;width:*500*;height:*50*" value=*"Test"*>Test

</div>

<div id=*"x"*>&nbsp; </div>

<div id=*"test1"* style="color:*white*;background-color:*red*; text-align:*left*;width:*500*;height:*50*" value=*"Test1"*>Test1

<marquee> This is another division</marquee>

<input type=*"submit"* value=*"update List"* onclick="updateFunction();">

</div>

<script>

**function** updateFunction()

{

//alert('updateFunction called');

**var** xmlhttp = **new** XMLHttpRequest();

//var url = "info.txt";

**var** url="http://localhost:8081/JSON-AJAX/SendDataServlet";

xmlhttp.open("GET", url, **true**);

xmlhttp.send();

xmlhttp.onreadystatechange = **function**() {

// status 200 ok

**if** (xmlhttp.readyState == 4 && xmlhttp.status == 200) {

**var** xx=xmlhttp.responseText;// array of JSON objects

//alert(xx);

**var** myArr = JSON.parse(xx);// array of JavaScript objects

**var** out = "<select>";

**var** i;

//alert("Array Length is "+myArr.length);

**for**(i = 0; i < myArr.length; i++) {

//myArr=JSON.parse(xx[i]);

out=out+"<option>"+myArr[i].name+"</option>";

}

out=out+"</select>";

//alert("out is "+out);

document.getElementById("test").innerHTML = out;

alert('successfully updated');

}

}

}

updateFunction();

</script>

</html>

**Step 2: Write the servlet:**

package com.ust\_global;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.json.simple.JSONObject;

/\*\*

\* Servlet implementation class SendDataServlet

\*/

public class SendDataServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public SendDataServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doPost(request,response);

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter pw=response.getWriter();

Object o[]=new Object[2];

JSONObject obj1=new JSONObject();

ArrayList<String> list=giveMeNames();

String json="[";

// Final output should be sent to AJAX

if (list.size()>0)

System.out.println("Greater than zero");

else

System.out.println("Less than or equal..."+list.size());

System.out.println("List 0 is "+list.get(0));

for (int i=0;i<list.size();i++){

obj1.put("name",(String)list.get(i));

json=json+obj1+",";

}

json=json.substring(0,json.length()-1);

json=json+"]";

pw.println(json);

}

/\*

\* http ready state:

0 Uninitialized - open() has not been called yet.

1 Loading - send() has not been called yet.

2 Loaded - send() has been called, headers and status are available.

3 Interactive - Downloading, responseText holds the partial data.

4 Completed - Finished with all operations.

Status:

200 completed

404 file not found

\*

\*/

public ArrayList<String> giveMeNames()

{

// Should not write here

ArrayList<String> list=new ArrayList<String>();

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","username","password");

if (con==null)

System.out.println("null connection");

else

System.out.println("not null connection");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from employee");

while(rs.next()) {

// System.out.println("rec..."+rs.getString(2));

list.add(rs.getString(3));

}

}

catch(Exception e)

{

System.out.println("db not connecting");

}

return list;

}

}