Web Services

- Web Service is a process or a function which operates over web.
- It may work in one of the following way
  - GET - send some parameter and get some results (Eg. Getting balance for an account number)
  - POST - send some parameter and it is consumed by the service (Eg. A logistics delivery person updates the delivery information in the server)
- Web Services are technology independent. It means web service can be created in one technology and consumed from different technology
- Types of Web services
  - SOAP
  - REST

Creating REST service in spring mvc

1) Create a table

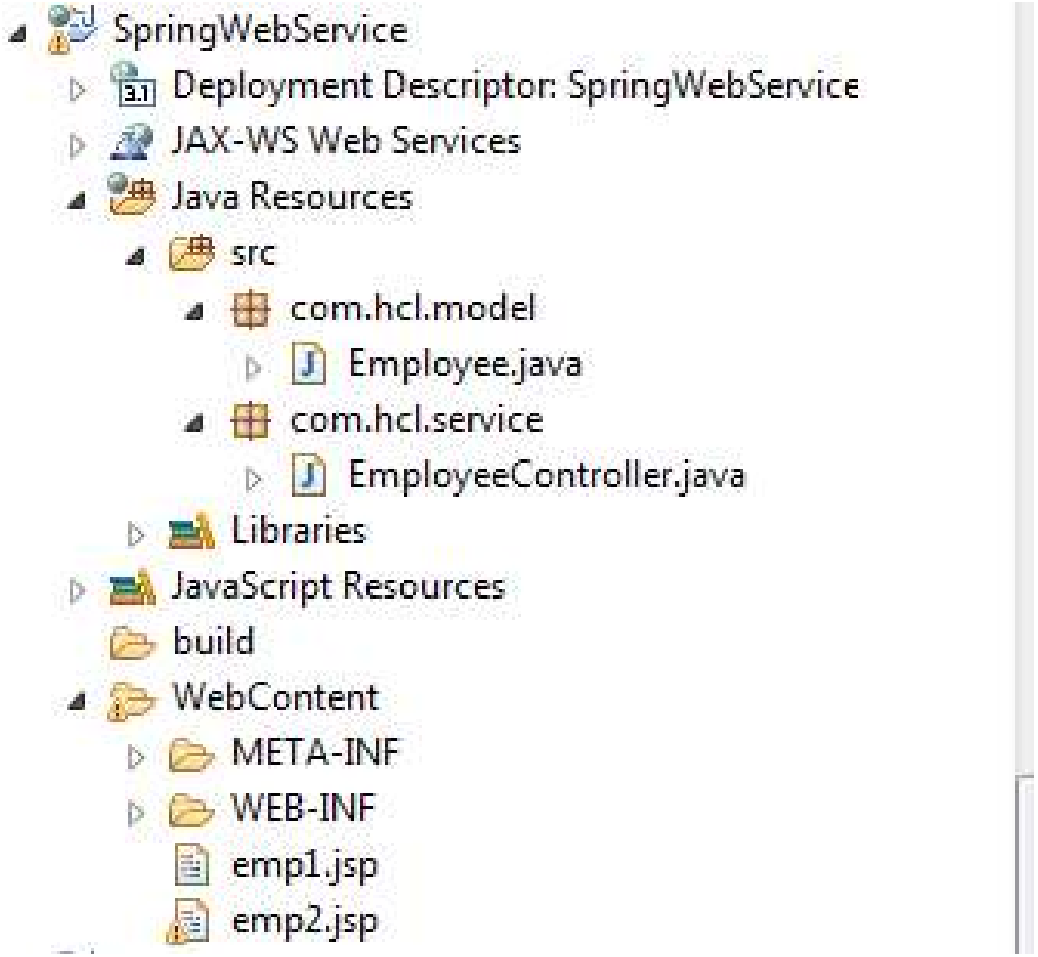
```
CREATE TABLE HCLEMP(EmpID VARCHAR2(3) PRIMARY KEY,
EmpName VARCHAR2(25), Desig VARCHAR2(15), City
VARCHAR2(20))
```

Creating REST service in spring mvc

1) Create a table with some records
2) Create a dynamic web project
3) In project, add
    1) spring jars
4) In Deployment, add
    1) spring jars
    2) ojdbc7.jar
    3) jackson jars (jackson-core, jackson-annotations & jackson-databind)

Creating REST service in spring mvc

File organization



Creating REST service in spring mvc

Employee.java (model class)

```
package com.hcl.model;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class Employee {
	String empID;
	String empName;
	String desig;
	String city;
	public String getEmpID() {
		return empID;
	}
	public void setEmpID(String empID) {
		this.empID = empID;
	}
	public String getEmpName() {
		return empName;
	}
```

Creating REST service in spring mvc

Employee.java
(Contd…)

```
public void setEmpName(String empName) {
        this.empName = empName;
}
public String getDesig() {
        return desig;
}
public void setDesig(String desig) {
        this.desig = desig;
}
public String getCity() {
        return city;
}
public void setCity(String city) {
        this.city = city;
}
```

Creating REST service in spring mvc

Employee.java
(Contd…)

```
public static Employee getEmployee(String empID) {
    Employee e = new Employee();
    String userid="dbuser";
    String password = "1234";
    String url = "jdbc:oracle:thin:@localhost:1521/XE";
    Connection con;
    Statement stmt;
    String qry;
    ResultSet rs;
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con = DriverManager.getConnection(url, userid, password);
        stmt= con.createStatement();
        qry="SELECT * FROM HclEmp WHERE EmpID='" + empID + "'";
        rs = stmt.executeQuery(qry);
        e.setEmpID(empID);
        if(rs.next()) {
            e.setEmpName(rs.getString("EmpName"));
            e.setDesig(rs.getString("Desig"));
            e.setCity(rs.getString("City"));
```

Creating REST service in spring mvc

Employee.java
(Contd…)

```
            } else {
                e.setEmpName("not found");
                e.setDesig("not found");
                e.setCity("not found");
            }
            stmt.close();
            con.close();
        }
        catch(java.lang.ClassNotFoundException ex) {
            System.out.println("Oracle Driver not found");
        }
        catch(SQLException ex) {
            System.err.println("SQLException: " + ex.getMessage());
        }
        return e;
    }
}
```

Creating REST service in spring mvc

EmployeeController.java
(RestController)

```
package com.hcl.service;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import com.hcl.model.Employee;
@RestController
public class EmployeeController {
	@RequestMapping(value = "/getemployee/{empID}", method = RequestMethod.GET)
	public Employee getEmployee(@PathVariable("empID") String empID) {
		Employee e =  Employee.getEmployee(empID);
		return e;
	}
}
```

Creating REST service in spring mvc

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"
  version="3.1">
  <display-name> SpringWebService </display-name>
  <servlet>
    <servlet-name>dispatcher</servlet-name>
    <servlet-class>
        org.springframework.web.servlet.DispatcherServlet
     </servlet-class>
     <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>dispatcher</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>
```

Creating REST service in spring mvc

dispatcher-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
  <beans xmlns:mvc="http://www.springframework.org/schema/mvc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
      http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
      http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd
      http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">
  <context:component-scan base-package="com.hcl" />
  <mvc:annotation-driven/>
 </beans>
```

Consuming REST service

REST Service can be consumed from….

1) JavaScript
2) Angular JS
3) Jquery (not discussed)

Consuming REST service using JavaScript

emp1.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>Employee Search</title>
</head>
<body>
      Emp ID: <input type='text' id='empID'/>
      <input type='button' value='get' onclick='getEmployee();'/> <br/>
      Name: <input type='text' id='empName' disabled/> <br/>
      Designation: <input type='text' id='desig' disabled/> <br/>
      City: <input type='text' id='city' disabled/> <br/>
```

Consuming REST service using JavaScript

emp1.jsp
Contd…

```
<script>
    function getEmployee() {
    var empID = document.getElementById("empID").value
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
            var emp = JSON.parse(this.responseText);
            document.getElementById("empName").value = emp.empName;
            document.getElementById("desig").value = emp.desig;
            document.getElementById("city").value = emp.city;
      }
     };
     xhttp.open("GET", "getemployee/" + empID , true);
     xhttp.send();
    }
</script>

</body>
</html>
```

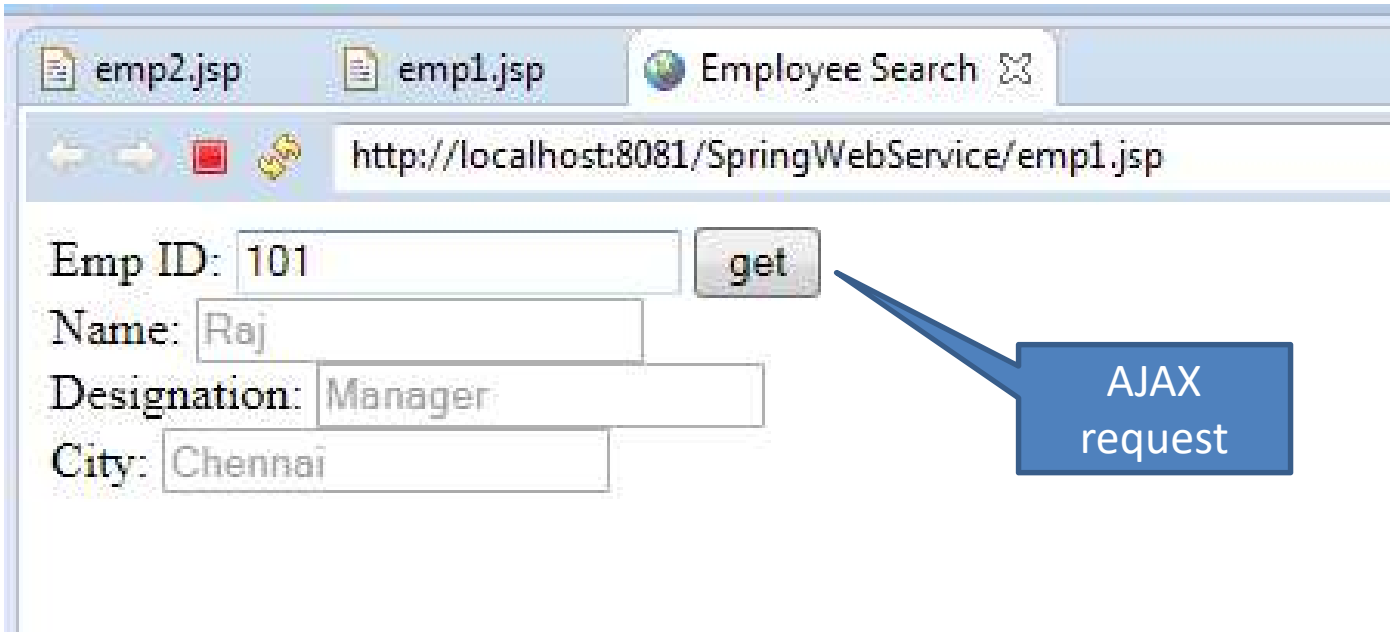
Consuming REST service using JavaScript

Ready state values

Value	State	Description
0	UNSENT	Client has been created. open() not called yet.
1	OPENED	open() has been called.
2	HEADERS_RECEIVED	send() has been called, and headers and status are available.
3	LOADING	Downloading; responseText holds partial data.
4	DONE	The operation is complete.

Consuming REST service using JavaScript

Run emp1.jsp



Consuming REST service using angular JS

emp2.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>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
<body>
<div ng-app="EmpApp" ng-controller="EmpController">
        Emp ID: <input type="text" ng-model="emp.empID">
        <button ng-click="getEmp()"> Search </button> <br/>
        Name: {{emp.empName}} <br/>
        Designation: {{emp.desig}} <br/>
        City: {{emp.city}} <br/>
</div>
<script>
```

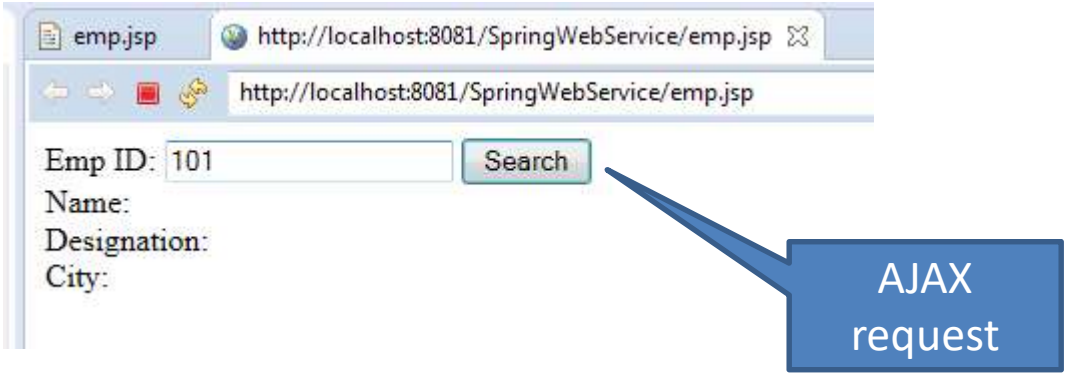
Consuming REST service using angular JS

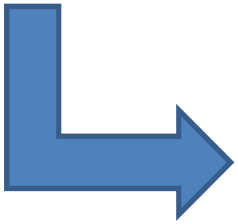
emp2.jsp
(Contd…)

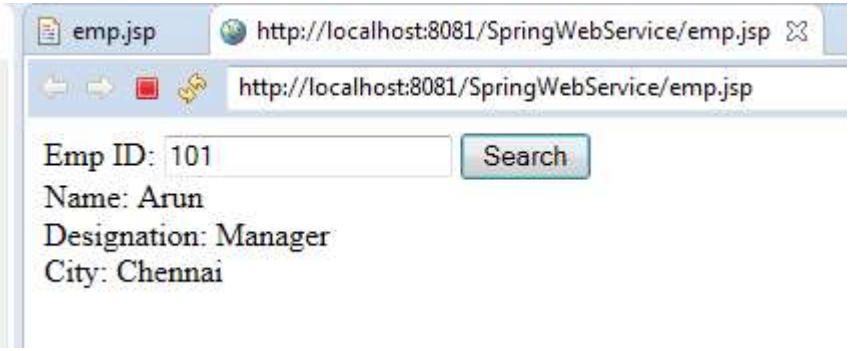
```
var app = angular.module('EmpApp', []);
app.controller('EmpController', function($scope,$http) {
   $scope.emp = [];
   $scope.getEmp = function() {
   $http({
         method : 'GET',
         url : 'getemployee/' + $scope.emp.empID
   }).then(function successCallback(response) {
         $scope.emp = response.data;
      }, function errorCallback(response) {
         alert("Data Error");
         console.log(response.statusText);
      });
   };
});
</script>
</body>
</html>
```

Consuming REST service using angular JS

Run emp.jsp







Consuming REST service using angular JS

For invalid id