**Problem Statement - Display Employee List and Edit Employee form using RESTful Web Service**   
   
In the previous angular module, we developed a screen that lists employees and it was populated with hard coded values. Now this angular application has be changed to get the data from RESTful Web Service developed in Spring. The following are the high level activities that needs to be done to accomplish this:

* Create static employee list data using spring xml configuration

* Create a REST Service that reads data from xml configuration and returns it

* Make changes in angular component to consume the created REST Service

Once above activities are completed, clicking on the Edit button against each employee should display Edit Employee form with values retrieved from RESTful Web Service. This will also involve activities similar to the one specified above.  
   
NOTE: There is no specific activity as part of this hands on, refer the next hands ons that covers above three activities in detail.

**Solution**

**Employee.java**

package com.cognizant.employeeservice.model;  
  
public class Employee {  
 private int id;  
 private String name;  
 private String department;  
 private double salary;  
  
 public Employee() {  
 super();  
 }  
 public Employee(int id, String name, String department, double salary) {  
 this.id = id;  
 this.name = name;  
 this.department = department;  
 this.salary = salary;  
 }  
 public int getId() {  
 return id;  
 }  
 public void setId(int id) {  
 this.id = id;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
 public String getDepartment() {  
 return department;  
 }  
 public void setDepartment(String department) {  
 this.department = department;  
 }  
 public double getSalary() {  
 return salary;  
 }  
 public void setSalary(double salary) {  
 this.salary = salary;  
 }  
}

**employeelist.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 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.xsd">  
  
 <bean id="employee1" class="com.cognizant.employeeservice.model.Employee">  
 <property name="id" value="1"/>  
 <property name="name" value="John Doe"/>  
 <property name="department" value="IT"/>  
 <property name="salary" value="50000"/>  
 </bean>  
  
 <bean id="employee2" class="com.cognizant.employeeservice.model.Employee">  
 <property name="id" value="2"/>  
 <property name="name" value="Jane Smith"/>  
 <property name="department" value="HR"/>  
 <property name="salary" value="45000"/>  
 </bean>  
  
 <bean id="employeeList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="employee1"/>  
 <ref bean="employee2"/>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

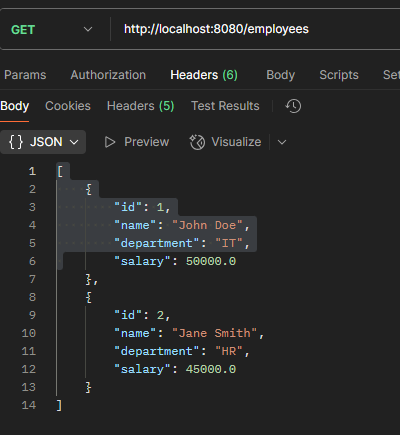
**EmployeeController.java**

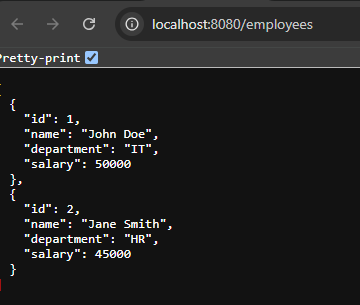
package com.cognizant.employeeservice.controller;  
  
import com.cognizant.employeeservice.model.Employee;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/employees")  
public class EmployeeController {  
  
 @Autowired  
 private ApplicationContext context;  
  
 @GetMapping  
 public List<Employee> getAllEmployees() {  
 return (List<Employee>) context.getBean("employeeList");  
 }  
}

**EmployeeWebServiceApplication.java**

package com.cognizant.employeeservice;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.annotation.ImportResource;  
  
@SpringBootApplication  
@ImportResource("classpath:employeelist.xml")  
public class EmployeeWebServiceApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(EmployeeWebServiceApplication.class, args);  
 }  
}

**Output**





**Create static employee list data using spring xml configuration**   
   
Follow steps below to accomplish this activity:

* Incorporate the following in employee.xml:
  + Create one or two more departments
  + Create four more instances of Employee. (use employee sample data from angular)
  + Reuse existing skills instead of creating new ones
  + Include all four employee instances in an ArrayList.

* In EmployeeDao, incorporate the following:
  + Create static variable with name EMPLOYEE\_LIST of type ArrayList<Employee>
  + Include constructor that reads employee list from xml config and set the EMPLOYEE\_LIST
  + Create method getAllEmployees() that returns the EMPLOYEE\_LIST

**Solution**

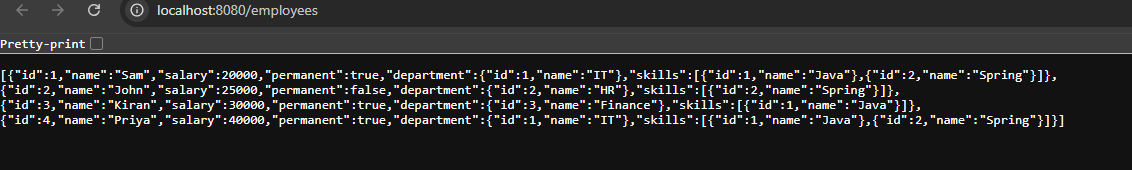
**employee.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:util="http://www.springframework.org/schema/util"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans  
 https://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/util  
 https://www.springframework.org/schema/util/spring-util.xsd">  
  
 <!-- Skills -->  
 <bean id="skillJava" class="com.cognizant.employeeservice.model.Skill">  
 <property name="id" value="1"/>  
 <property name="name" value="Java"/>  
 </bean>  
  
 <bean id="skillSpring" class="com.cognizant.employeeservice.model.Skill">  
 <property name="id" value="2"/>  
 <property name="name" value="Spring"/>  
 </bean>  
  
 <!-- Departments -->  
 <bean id="departmentIT" class="com.cognizant.employeeservice.model.Department">  
 <property name="id" value="1"/>  
 <property name="name" value="IT"/>  
 </bean>  
  
 <bean id="departmentHR" class="com.cognizant.employeeservice.model.Department">  
 <property name="id" value="2"/>  
 <property name="name" value="HR"/>  
 </bean>  
  
 <bean id="departmentFinance" class="com.cognizant.employeeservice.model.Department">  
 <property name="id" value="3"/>  
 <property name="name" value="Finance"/>  
 </bean>  
  
 <!-- Employees -->  
 <bean id="employee1" class="com.cognizant.employeeservice.model.Employee">  
 <property name="id" value="1"/>  
 <property name="name" value="Sam"/>  
 <property name="salary" value="20000"/>  
 <property name="permanent" value="true"/>  
 <property name="department" ref="departmentIT"/>  
 <property name="skills">  
 <list>  
 <ref bean="skillJava"/>  
 <ref bean="skillSpring"/>  
 </list>  
 </property>  
 </bean>  
  
 <bean id="employee2" class="com.cognizant.employeeservice.model.Employee">  
 <property name="id" value="2"/>  
 <property name="name" value="John"/>  
 <property name="salary" value="25000"/>  
 <property name="permanent" value="false"/>  
 <property name="department" ref="departmentHR"/>  
 <property name="skills">  
 <list>  
 <ref bean="skillSpring"/>  
 </list>  
 </property>  
 </bean>  
  
 <bean id="employee3" class="com.cognizant.employeeservice.model.Employee">  
 <property name="id" value="3"/>  
 <property name="name" value="Kiran"/>  
 <property name="salary" value="30000"/>  
 <property name="permanent" value="true"/>  
 <property name="department" ref="departmentFinance"/>  
 <property name="skills">  
 <list>  
 <ref bean="skillJava"/>  
 </list>  
 </property>  
 </bean>  
  
 <bean id="employee4" class="com.cognizant.employeeservice.model.Employee">  
 <property name="id" value="4"/>  
 <property name="name" value="Priya"/>  
 <property name="salary" value="40000"/>  
 <property name="permanent" value="true"/>  
 <property name="department" ref="departmentIT"/>  
 <property name="skills">  
 <list>  
 <ref bean="skillJava"/>  
 <ref bean="skillSpring"/>  
 </list>  
 </property>  
 </bean>  
  
 <!-- Employee List -->  
 <util:list id="employeeList" list-class="java.util.ArrayList">  
 <ref bean="employee1"/>  
 <ref bean="employee2"/>  
 <ref bean="employee3"/>  
 <ref bean="employee4"/>  
 </util:list>  
  
</beans>

**EmployeeDao.java**

package com.cognizant.employeeservice.dao;  
  
import com.cognizant.employeeservice.model.Employee;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Repository  
public class EmployeeDao {  
  
 public static List<Employee> EMPLOYEE\_LIST;  
  
 public EmployeeDao() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");  
 EMPLOYEE\_LIST = (List<Employee>) context.getBean("employeeList");  
 }  
  
 public List<Employee> getAllEmployees() {  
 return EMPLOYEE\_LIST;  
 }  
}

**Output**



**Create REST service to gets all employees**   
   
Follow steps below to accomplish this activity:

* In EmployeeService, incorporate the following:
  + Change the annotation for this class from @Component to @Service
  + Create method getAllEmployees() that invokes employeeDao.getAllEmployees() and return the employee list
  + Define @Transactional annotation for this method.

* In EmployeeController, incorporate the following:
  + Include a new get method with name getAllEmployees() that returns the employee list
  + Mark this method as GetMapping annotation with the URL as '/employees'
  + Within this method invoke employeeService.getAllEmployees() and return the same.
* Test the service using postman.

**Solution**

**EmployeeService.java**

package com.cognizant.employeeservice.service;  
  
import com.cognizant.employeeservice.dao.EmployeeDao;  
import com.cognizant.employeeservice.model.Employee;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
  
  
import java.util.List;  
  
@Service  
public class EmployeeService {  
  
 @Autowired  
 private EmployeeDao employeeDao;  
  
 @Transactional  
 public List<Employee> getAllEmployees() {  
 return employeeDao.getAllEmployees();  
 }  
}

**EmployeeController.java**

package com.cognizant.employeeservice.controller;  
  
import com.cognizant.employeeservice.dao.EmployeeDao;  
import com.cognizant.employeeservice.model.Employee;  
import com.cognizant.employeeservice.service.EmployeeService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
public class EmployeeController {  
 @Autowired  
 private EmployeeService employeeService;  
 @GetMapping("/employees")  
 public List<Employee> getAllEmployees() {  
 return employeeService.getAllEmployees(); // call to service, not DAO directly  
 }  
}

**Outptut**

[

{

"id": 1,

"name": "Sam",

"salary": 20000.0,

"permanent": true,

"department": {

"id": 1,

"name": "IT"

},

"skills": [

{

"id": 1,

"name": "Java"

},

{

"id": 2,

"name": "Spring"

}

]

},

{

"id": 2,

"name": "John",

"salary": 25000.0,

"permanent": false,

"department": {

"id": 2,

"name": "HR"

},

"skills": [

{

"id": 2,

"name": "Spring"

}

]

},

{

"id": 3,

"name": "Kiran",

"salary": 30000.0,

"permanent": true,

"department": {

"id": 3,

"name": "Finance"

},

"skills": [

{

"id": 1,

"name": "Java"

}

]

},

{

"id": 4,

"name": "Priya",

"salary": 40000.0,

"permanent": true,

"department": {

"id": 1,

"name": "IT"

},

"skills": [

{

"id": 1,

"name": "Java"

},

{

"id": 2,

"name": "Spring"

}

]

}

]

**Create REST service for department**

Create a new service to get all the departments.

Follow steps below to achieve this:

* Create a new REST Service, define below list of classes and respective methods:
  + DepartmentController
    - getAllDepartments() with URL "/departments", this method will return array of departments
  + DepartmentService
    - getAllDepartments()
  + DepartmentDao
    - getAllDepartments() - Create a static variable DEPARTMENT\_LIST, this should be populated from spring xml configuration

* Test the service using postman.
* Also verify if department REST service is called by looking into the logs.

**Solution**

**DepartmentDao.java**

package com.cognizant.employeeservice.dao;  
  
import com.cognizant.employeeservice.model.Department;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Repository  
public class DepartmentDao {  
 private static List<Department> DEPARTMENT\_LIST;  
 @Autowired  
 public DepartmentDao(List<Department> departmentList) {  
 DEPARTMENT\_LIST = departmentList;  
 }  
 public List<Department> getAllDepartments() {  
 return DEPARTMENT\_LIST;  
 }  
}

**DepartmentService.java**

package com.cognizant.employeeservice.service;  
  
import com.cognizant.employeeservice.dao.DepartmentDao;  
import com.cognizant.employeeservice.model.Department;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
  
import java.util.List;  
  
@Service  
public class DepartmentService {  
 @Autowired  
 private DepartmentDao departmentDao;  
 @Transactional  
 public List<Department> getAllDepartments() {  
 return departmentDao.getAllDepartments();  
 }  
}  
**DepartmentController.java**

package com.cognizant.employeeservice.controller;  
  
import com.cognizant.employeeservice.model.Department;  
import com.cognizant.employeeservice.service.DepartmentService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
public class DepartmentController {  
 @Autowired  
 private DepartmentService departmentService;  
 @GetMapping("/departments")  
 public List<Department> getAllDepartments() {  
 System.out.println("Department REST service called...");  
 return departmentService.getAllDepartments();  
 }  
}

**Output**

