**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.

**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

**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.

**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.

**EmployeeController.java**

package com.example.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.example.dao.EmployeeDao;

import com.example.model.Employee;

@RestController

@RequestMapping("/api")

@CrossOrigin(origins = "\*") // Allow Angular to access this

public class EmployeeController {

    @Autowired

    private EmployeeDao employeeDao;

    @GetMapping("/employees")

    public List<Employee> getAllEmployees() {

        return employeeDao.getAllEmployees();

    }

    // Optional: add this to test

    @GetMapping("/test")

    public String test() {

        return "API is working!";

    }

}

**EmployeeDao.java**

package com.example.dao;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Repository;

import com.example.model.Employee;

@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;

    }

}

**Department.java**

**package com.example.model;**

**public class Department {**

**private int id;**

**private String name;**

**public Department() {}**

**public Department(int id, String name) {**

**this.id = id;**

**this.name = name;**

**}**

**// Getters and Setters**

**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;**

**}**

**}**

**Skill.java**

**package com.example.model;**

**public class Skill {**

**private int id;**

**private String name;**

**public Skill() {**

**}**

**public int getId() {**

**return id;**

**}**

**public void setId(int id) {  // ⚠️ REQUIRED by Spring XML**

**this.id = id;**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {  // ⚠️ REQUIRED by Spring XML**

**this.name = name;**

**}**

**@Override**

**public String toString() {**

**return "Skill{" +**

**"id=" + id +**

**", name='" + name + '\'' +**

**'}';**

**}**

**}**

**Employee.java**

**package com.example.model;**

**import java.util.List;**

**public class Employee {**

**private int id;**

**private String name;**

**private String designation;**

**private double salary;**

**private Department department;**

**private List<Skill> skills;**

**public Employee() {}**

**public Employee(int id, String name, String designation, double salary, Department department, List<Skill> skills) {**

**this.id = id;**

**this.name = name;**

**this.designation = designation;**

**this.salary = salary;**

**this.department = department;**

**this.skills = skills;**

**}**

**// Getters and Setters**

**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 getDesignation() {**

**return designation;**

**}**

**public void setDesignation(String designation) {**

**this.designation = designation;**

**}**

**public double getSalary() {**

**return salary;**

**}**

**public void setSalary(double salary) {**

**this.salary = salary;**

**}**

**public Department getDepartment() {**

**return department;**

**}**

**public void setDepartment(Department department) {**

**this.department = department;**

**}**

**public List<Skill> getSkills() {**

**return skills;**

**}**

**public void setSkills(List<Skill> skills) {**

**this.skills = skills;**

**}**

**}**

**DemoApplication.java**

**package com.example;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**import org.springframework.context.annotation.ImportResource;**

**@SpringBootApplication**

**@ImportResource("classpath:employee.xml")**

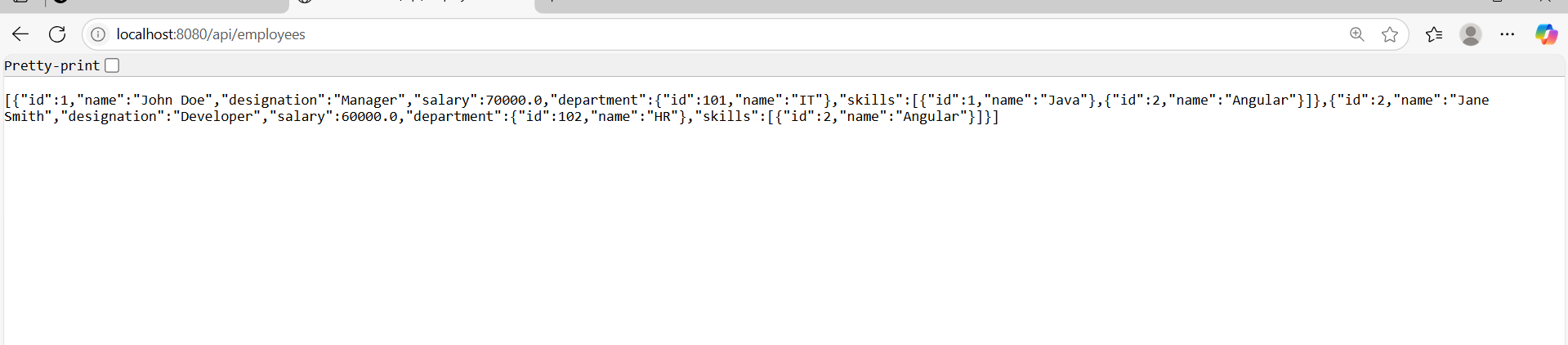
**public class DemoApplication {**

**public static void main(String[] args) {**

**SpringApplication.run(DemoApplication.class, args);**

**}**

**}**

****