**Hands on 2: 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: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">

<bean id="employeeList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.cognizant.spring\_learn.model.Employee">

<property name="id" value="1"/>

<property name="name" value="John"/>

<property name="salary" value="45000"/>

<property name="permanent" value="true"/>

</bean>

<bean class="com.cognizant.spring\_learn.model.Employee">

<property name="id" value="2"/>

<property name="name" value="Jane"/>

<property name="salary" value="55000"/>

<property name="permanent" value="false"/>

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**Employee.java**

package com.cognizant.spring\_learn.model;  
  
public class Employee {  
 private int id;  
 private String name;  
 private double salary;  
 private boolean permanent;  
  
 // Constructors  
 public Employee() {}  
 public Employee(int id, String name, double salary,boolean permanent) {  
 this.id = id;  
 this.name = name;  
 this.salary = salary;  
 this.permanent=permanent;  
  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setSalary(double salary) {  
 this.salary = salary;  
 }  
  
 public double getSalary() {  
 return salary;  
 }  
  
 public void setPermanent(boolean permanent) {  
 this.permanent = permanent;  
 }  
  
 public boolean isPermanent() {  
 return permanent;  
 }  
}

**EmployeeDao.java**

package com.cognizant.spring\_learn.dao;

import com.cognizant.spring\_learn.model.Employee;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

public class EmployeeDao {

private 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;

}

}

**EmployeeController.java:**

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.dao.EmployeeDao;

import com.cognizant.spring\_learn.model.Employee;

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

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

import java.util.List;

@RestController

public class EmployeeController {

private final EmployeeDao employeeDao;

public EmployeeController() {

this.employeeDao = new EmployeeDao(); // Load XML

}

@GetMapping("/employees")

public List<Employee> getEmployees() {

return employeeDao.getAllEmployees();

}

}

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

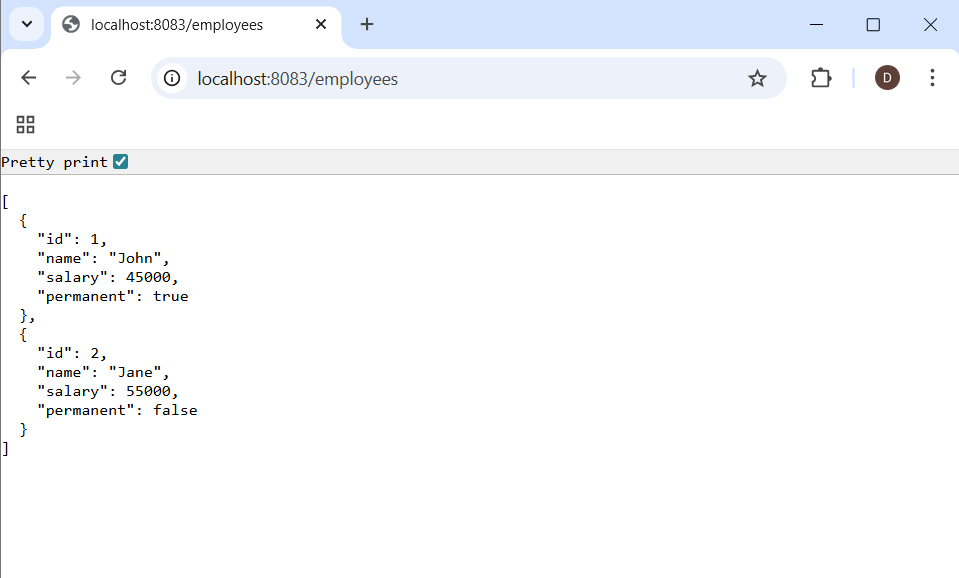
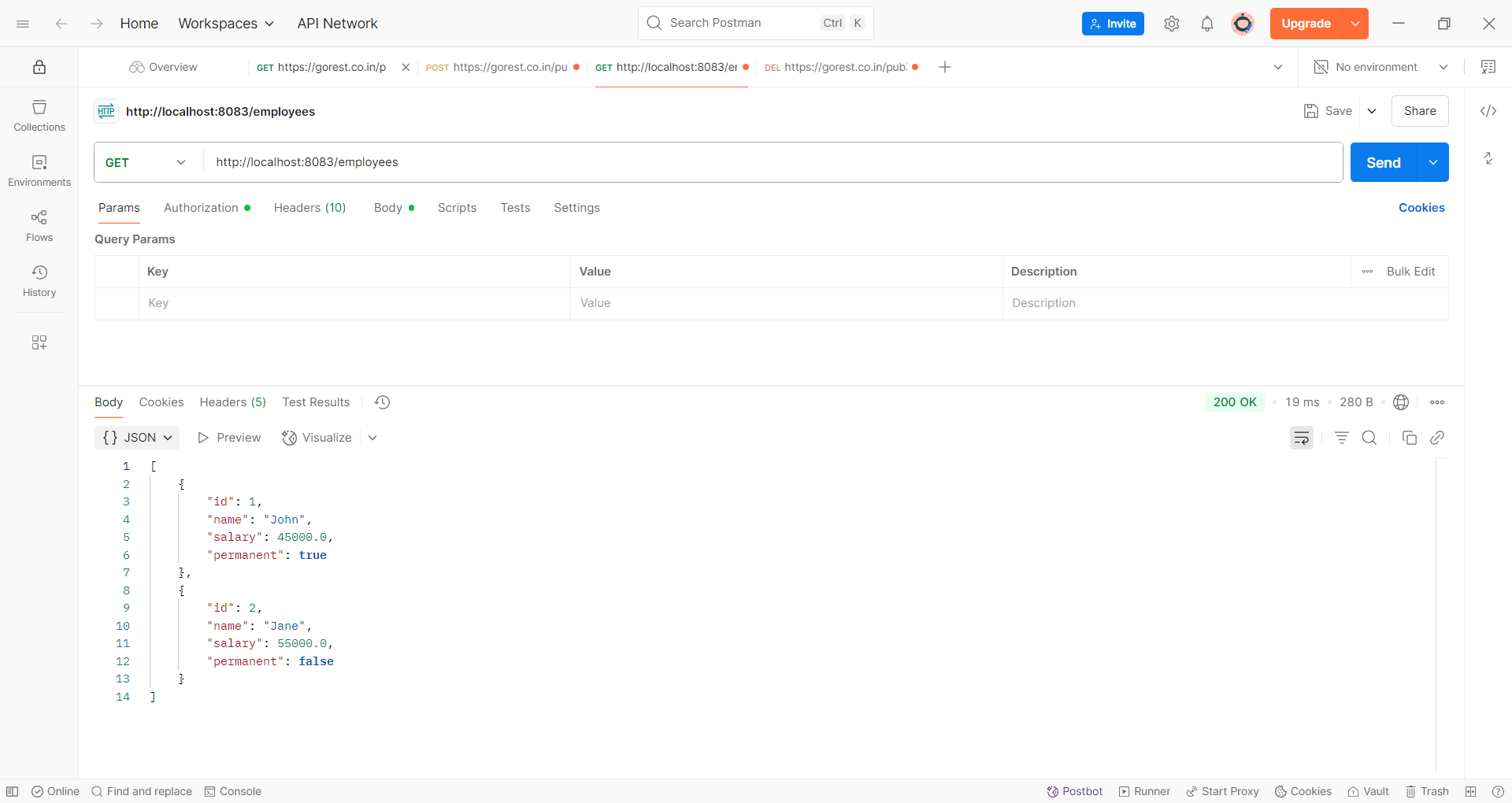
public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

**Output:**

****