**WEEK 4: ADDITIONAL HANDS-ON**

**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:

**1. Create Static Employee List in Spring 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

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/util

http://www.springframework.org/schema/util/spring-util.xsd">

<util:list id="employeeList">

<bean class="com.cognizant.springlearn.model.Employee">

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

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

<property name="salary" value="5000"/>

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

</bean>

<bean class="com.cognizant.springlearn.model.Employee">

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

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

<property name="salary" value="7000"/>

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

</bean>

</util:list>

</beans>

**2. Create Employee Model Class**

package com.cognizant.springlearn.model;

public class Employee {

private int id;

private String name;

private double salary;

private boolean permanent;

public Employee() {

System.out.println("Inside Employee Constructor");

}

public int getId() {

System.out.println("Getting ID");

return id;

}

public void setId(int id) {

System.out.println("Setting ID");

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

System.out.println("Setting Name");

this.name = name;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

System.out.println("Setting Salary");

this.salary = salary;

}

public boolean isPermanent() {

return permanent;

}

public void setPermanent(boolean permanent) {

this.permanent = permanent;

}

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent + "]";

}

}

**3. Create REST Controller**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Employee;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

public class EmployeeController {

ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");

@GetMapping("/employees")

public List<Employee> getAllEmployees() {

List<Employee> employeeList = (List<Employee>) context.getBean("employeeList");

return employeeList;

}

@GetMapping("/employees/{id}")

public Employee getEmployeeById(@PathVariable int id) {

List<Employee> employeeList = (List<Employee>) context.getBean("employeeList");

return employeeList.stream()

.filter(emp -> emp.getId() == id)

.findFirst()

.orElse(null);

}

}

**4. Update application.properties**

server.port=8083

**5. Update Angular Code to Consume the REST API**

getEmployees(): Observable<Employee[]> {

return this.http.get<Employee[]>('http://localhost:8083/employees');

}

getEmployeeById(id: number): Observable<Employee> {

return this.http.get<Employee>('http://localhost:8083/employees/' + id);

}

**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

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/util

http://www.springframework.org/schema/util/spring-util.xsd">

<bean id="dept1" class="com.cognizant.springlearn.model.Department">

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

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

</bean>

<bean id="dept2" class="com.cognizant.springlearn.model.Department">

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

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

</bean>

<bean id="skill1" class="com.cognizant.springlearn.model.Skill">

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

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

</bean>

<bean id="skill2" class="com.cognizant.springlearn.model.Skill">

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

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

</bean>

<util:list id="employeeList">

<bean class="com.cognizant.springlearn.model.Employee">

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

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

<property name="salary" value="50000" />

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

<property name="department" ref="dept1" />

<property name="skillList">

<list>

<ref bean="skill1" />

<ref bean="skill2" />

</list>

</property>

</bean>

<bean class="com.cognizant.springlearn.model.Employee">

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

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

<property name="salary" value="60000" />

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

<property name="department" ref="dept2" />

<property name="skillList">

<list>

<ref bean="skill1" />

</list>

</property>

</bean>

<bean class="com.cognizant.springlearn.model.Employee">

<property name="id" value="3" />

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

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

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

<property name="department" ref="dept1" />

<property name="skillList">

<list>

<ref bean="skill2" />

</list>

</property>

</bean>

<bean class="com.cognizant.springlearn.model.Employee">

<property name="id" value="4" />

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

<property name="salary" value="65000" />

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

<property name="department" ref="dept2" />

<property name="skillList">

<list>

<ref bean="skill1" />

<ref bean="skill2" />

</list>

</property>

</bean>

</util:list>

</beans>

**2. Create EmployeeDao Class**

package com.cognizant.springlearn.dao;

import com.cognizant.springlearn.model.Employee;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.ArrayList;

public class EmployeeDao {

public static ArrayList<Employee> EMPLOYEE\_LIST;

public EmployeeDao() {

ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");

EMPLOYEE\_LIST = (ArrayList<Employee>) context.getBean("employeeList");

}

public ArrayList<Employee> getAllEmployees() {

return 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.

Solution:

**EmployeeService.java**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.dao.EmployeeDao;

import com.cognizant.springlearn.model.Employee;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import java.util.ArrayList;

@Service // Changed from @Component to @Service

public class EmployeeService {

@Transactional

public ArrayList<Employee> getAllEmployees() {

EmployeeDao employeeDao = new EmployeeDao();

return employeeDao.getAllEmployees();

}

}

**2. EmployeeController.java**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Employee;

import com.cognizant.springlearn.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();

}

}

**Test the REST API**

[

{

"id": 1,

"name": "John",

"salary": 50000.0,

"permanent": true,

"department": {

"id": 1,

"name": "HR"

},

"skillList": [

{ "id": 1, "name": "Java" },

{ "id": 2, "name": "Angular" }

]

},

{

"id": 2,

"name": "Jane",

"salary": 60000.0,

"permanent": false,

"department": {

"id": 2,

"name": "Finance"

},

"skillList": [

{ "id": 1, "name": "Java" }

]

}

]

**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:

**department.xml – Spring Configuration**

<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="department1" class="com.cognizant.springlearn.model.Department">

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

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

</bean>

<bean id="department2" class="com.cognizant.springlearn.model.Department">

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

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

</bean>

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

<constructor-arg>

<list>

<ref bean="department1"/>

<ref bean="department2"/>

</list>

</constructor-arg>

</bean>

</beans>

**2. DepartmentDao.java**

package com.cognizant.springlearn.dao;

import com.cognizant.springlearn.model.Department;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

public class DepartmentDao {

private static List<Department> DEPARTMENT\_LIST;

public DepartmentDao() {

ApplicationContext context = new ClassPathXmlApplicationContext("department.xml");

DEPARTMENT\_LIST = (List<Department>) context.getBean("departmentList");

}

public List<Department> getAllDepartments() {

return DEPARTMENT\_LIST;

}

}

**3. DepartmentService.java**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.dao.DepartmentDao;

import com.cognizant.springlearn.model.Department;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Service

public class DepartmentService {

private DepartmentDao departmentDao = new DepartmentDao();

@Transactional

public List<Department> getAllDepartments() {

return departmentDao.getAllDepartments();

}

}

**4. DepartmentController.java**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Department;

import com.cognizant.springlearn.service.DepartmentService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

public class DepartmentController {

private static final Logger LOGGER = LoggerFactory.getLogger(DepartmentController.class);

@Autowired

private DepartmentService departmentService;

@GetMapping("/departments")

public List<Department> getAllDepartments() {

LOGGER.info("START getAllDepartments()");

List<Department> departments = departmentService.getAllDepartments();

LOGGER.info("END getAllDepartments()");

return departments;

}

}

**Testing with Postman**

[

{

"id": 1,

"name": "HR"

},

{

"id": 2,

"name": "Finance"

}

]