**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.

**Source Code:**

**employee.xml:**

<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.springlearn.model.Employee">

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

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

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

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

</bean>

<bean id="employee2" class="com.cognizant.springlearn.model.Employee">

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

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

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

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

</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.springlearn.controller;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

import com.cognizant.springlearn.model.Employee;

@RestController

public class EmployeeController {

@GetMapping("/employees")

public List<Employee> getAllEmployees() {

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

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

return employeeList;

}

}

**Employee.java:**

Path: src/main/java/com/cognizant/springlearn/model/Employee.java

package com.cognizant.springlearn.model;

import java.util.Date;

public class Employee {

private int id;

private String name;

private double salary;

private boolean permanent;

private Date dateOfBirth;

public Employee() {

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

}

public int getId() {

return id;

}

public void setId(int id) {

System.out.println("Inside setId()");

this.id = id;

}

public String getName()

return name;

}

public void setName(String name) {

System.out.println("Inside setName()");

this.name = name;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

System.out.println("Inside setSalary()");

this.salary = salary;

}

public boolean isPermanent() {

return permanent;

}

public void setPermanent(boolean permanent) {

System.out.println("Inside setPermanent()");

this.permanent = permanent;

}

public Date getDateOfBirth() {

return dateOfBirth;

}

public void setDateOfBirth(Date dateOfBirth) {

System.out.println("Inside setDateOfBirth()");

this.dateOfBirth = dateOfBirth;

}

@Override

public String toString() {

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

+ ", dateOfBirth=" + dateOfBirth + "]";

}

}

**employee.service.ts:**

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { Observable } from 'rxjs';

export interface Employee {

id: number;

name: string;

salary: number;

permanent: boolean;

dateOfBirth: string;

}

@Injectable({

providedIn: 'root'

})

export class EmployeeService {

private apiUrl = 'http://localhost:8083/employees';

constructor(private http: HttpClient) {}

getAllEmployees(): Observable<Employee[]> {

return this.http.get<Employee[]>(this.apiUrl);

}

}

**employee-list.component.ts:**

import { Component, OnInit } from '@angular/core';

import { Employee, EmployeeService } from '../../services/employee.service';

@Component({

selector: 'app-employee-list',

templateUrl: './employee-list.component.html'

})

export class EmployeeListComponent implements OnInit {

employees: Employee[] = [];

constructor(private employeeService: EmployeeService) {}

ngOnInit(): void {

this.employeeService.getAllEmployees().subscribe(data => {

this.employees = data;

});

}

editEmployee(employee: Employee): void {

console.log('Editing:', employee);

}

}

**employee-list.component.html:**

<table>

<tr>

<th>ID</th><th>Name</th><th>Salary</th><th>Permanent</th><th>DOB</th><th>Action</th>

</tr>

<tr \*ngFor="let emp of employees">

<td>{{ emp.id }}</td>

<td>{{ emp.name }}</td>

<td>{{ emp.salary }}</td>

<td>{{ emp.permanent }}</td>

<td>{{ emp.dateOfBirth }}</td>

<td><button (click)="editEmployee(emp)">Edit</button></td>

</tr>

</table>

**OUTPUT:**

INFO ... SpringLearnApplication - Started SpringLearnApplication

INFO ... EmployeeController - Inside Employee Controller

Browser

ID | Name | Salary | Permanent | DOB | Action

-------------------------------------------------------------

1 | John Doe | 50000 | true | 1990-01-01| Edit

2 | Jane Smith | 55000 | false | 1992-05-05| Edit