**Problem Statement - Display Employee List and Edit Employee form using RESTful Web Service**

**Employee.java**

package com.example.model;

public class Employee {

private int id;

private String name;

private String department;

private double salary;

// Constructors

public Employee() {}

public Employee(int id, String name, String department, double salary) {

this.id = id;

this.name = name;

this.department = department;

this.salary = salary;

}

// 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 getDepartment() { return department; }

public void setDepartment(String department) { this.department = department; }

public double getSalary() { return salary; }

public void setSalary(double salary) { this.salary = salary; }

}

**employee-config.xml**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:context="http://www.springframework.org/schema/context"

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/context

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

<bean id="employee1" class="com.example.model.Employee">

<constructor-arg value="1"/>

<constructor-arg value="Alice"/>

<constructor-arg value="HR"/>

<constructor-arg value="55000"/>

</bean>

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

<constructor-arg value="2"/>

<constructor-arg value="Bob"/>

<constructor-arg value="IT"/>

<constructor-arg value="65000"/>

</bean>

<bean id="employee3" class="com.example.model.Employee">

<constructor-arg value="3"/>

<constructor-arg value="Charlie"/>

<constructor-arg value="Finance"/>

<constructor-arg value="70000"/>

</bean>

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

<constructor-arg>

<list>

<ref bean="employee1"/>

<ref bean="employee2"/>

<ref bean="employee3"/>

</list>

</constructor-arg>

</bean>

</beans>

**EmployeeService.java**

package com.example.service;

import com.example.model.Employee;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class EmployeeService {

private final List<Employee> employeeList;

public EmployeeService() {

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

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

}

public List<Employee> getAllEmployees() {

return employeeList;

}

public Employee getEmployeeById(int id) {

return employeeList.stream()

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

.findFirst()

.orElse(null);

}

}

**EmployeeController.java**

package com.example.controller;

import com.example.model.Employee;

import com.example.service.EmployeeService;

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

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

import java.util.List;

@RestController

@RequestMapping("/api/employees")

@CrossOrigin(origins = "http://localhost:4200") // Enable CORS for Angular

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@GetMapping

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

@GetMapping("/{id}")

public Employee getEmployeeById(@PathVariable int id) {

return employeeService.getEmployeeById(id);

}

}

**EmployeeApplication.java**

package com.example;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class EmployeeApplication {

public static void main(String[] args) {

SpringApplication.run(EmployeeApplication.class, args);

}

}

**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;

department: string;

salary: number;

}

@Injectable({

providedIn: 'root'

})

export class EmployeeService {

private apiUrl = 'http://localhost:8080/api/employees';

constructor(private http: HttpClient) {}

getEmployees(): Observable<Employee[]> {

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

}

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

return this.http.get<Employee>(`${this.apiUrl}/${id}`);

}

}

**employee-list.component.ts**

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

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

import { Router } from '@angular/router';

@Component({

selector: 'app-employee-list',

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

})

export class EmployeeListComponent implements OnInit {

employees: Employee[] = [];

constructor(private service: EmployeeService, private router: Router) {}

ngOnInit(): void {

this.service.getEmployees().subscribe(data => this.employees = data);

}

editEmployee(id: number) {

this.router.navigate(['/edit', id]);

}

}

**employee-list.component.html**

<h2>Employee List</h2>

<table border="1">

<tr>

<th>ID</th><th>Name</th><th>Department</th><th>Salary</th><th>Actions</th>

</tr>

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

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

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

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

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

<td>

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

</td>

</tr>

</table>

**edit-employee.component.ts**

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

import { ActivatedRoute } from '@angular/router';

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

@Component({

selector: 'app-edit-employee',

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

})

export class EditEmployeeComponent implements OnInit {

employee: Employee = { id: 0, name: '', department: '', salary: 0 };

constructor(

private route: ActivatedRoute,

private service: EmployeeService

) {}

ngOnInit(): void {

const id = Number(this.route.snapshot.paramMap.get('id'));

this.service.getEmployeeById(id).subscribe(data => this.employee = data);

}

}

**edit-employee.component.html**

<h2>Edit Employee</h2>

<div \*ngIf="employee">

<form>

<label>ID:</label> <input [(ngModel)]="employee.id" name="id" disabled /><br/>

<label>Name:</label> <input [(ngModel)]="employee.name" name="name" /><br/>

<label>Department:</label> <input [(ngModel)]="employee.department" name="department" /><br/>

<label>Salary:</label> <input [(ngModel)]="employee.salary" name="salary" /><br/>

<button>Update</button>

</form>

</div>

**app-routing.module.ts**

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

import { RouterModule, Routes } from '@angular/router';

import { EmployeeListComponent } from './employee-list/employee-list.component';

import { EditEmployeeComponent } from './edit-employee/edit-employee.component';

const routes: Routes = [

{ path: '', component: EmployeeListComponent },

{ path: 'edit/:id', component: EditEmployeeComponent }

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

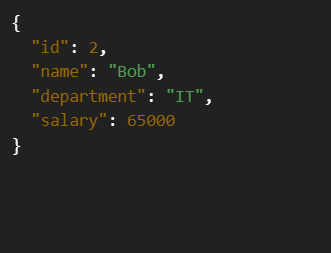
})

export class AppRoutingModule {}

**Output:**







**Problem Statement - Display Employee List and Edit Employee form using RESTful Web Service**

**employee-config.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="employeeList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.example.model.Employee">

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

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

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

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

</bean>

<bean class="com.example.model.Employee">

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

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

<property name="department" value="IT"/>

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

</bean>

<bean class="com.example.model.Employee">

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

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

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

<property name="salary" value="70000"/>

</bean>

</list>

</constructor-arg>

</bean>

</beans>

**Employee.java**

package com.example.model;

public class Employee {

private int id;

private String name;

private String department;

private double salary;

// Getters and Setters

}

**EmployeeService.java**

package com.example.service;

import com.example.model.Employee;

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

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private ApplicationContext context;

public List<Employee> getAllEmployees() {

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

}

public Employee getEmployeeById(int id) {

return getAllEmployees().stream()

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

.findFirst()

.orElse(null);

}

}

**EmployeeController.java**

package com.example.controller;

import com.example.model.Employee;

import com.example.service.EmployeeService;

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

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

import java.util.List;

@RestController

@RequestMapping("/api/employees")

@CrossOrigin(origins = "\*") // Enable CORS for Angular

public class EmployeeController {

@Autowired

private EmployeeService service;

@GetMapping

public List<Employee> getAllEmployees() {

return service.getAllEmployees();

}

@GetMapping("/{id}")

public Employee getEmployeeById(@PathVariable int id) {

return service.getEmployeeById(id);

}

}

**Application.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-config.xml") // Load XML bean

public class Application {

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

}

**employee.model.ts**

export interface Employee {

id: number;

name: string;

department: string;

salary: number;

}

**employee.service.ts**

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

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

import { Employee } from './employee.model';

import { Observable } from 'rxjs';

@Injectable({

providedIn: 'root',

})

export class EmployeeService {

private baseUrl = 'http://localhost:8080/api/employees';

constructor(private http: HttpClient) {}

getEmployees(): Observable<Employee[]> {

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

}

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

return this.http.get<Employee>(`${this.baseUrl}/${id}`);

}

}

**employee-list.component.ts**

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

import { Router } from '@angular/router';

import { Employee } from '../employee.model';

import { EmployeeService } from '../employee.service';

@Component({

selector: 'app-employee-list',

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

})

export class EmployeeListComponent implements OnInit {

employees: Employee[] = [];

constructor(private service: EmployeeService, private router: Router) {}

ngOnInit(): void {

this.service.getEmployees().subscribe((data) => {

this.employees = data;

});

}

edit(id: number) {

this.router.navigate(['/edit', id]);

}

}

**employee-list.component.html**

<table>

<thead>

<tr>

<th>ID</th><th>Name</th><th>Department</th><th>Salary</th><th>Action</th>

</tr>

</thead>

<tbody>

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

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

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

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

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

<td><button (click)="edit(emp.id)">Edit</button></td>

</tr>

</tbody>

</table>

**employee-edit.component.ts**

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

import { ActivatedRoute } from '@angular/router';

import { EmployeeService } from '../employee.service';

import { Employee } from '../employee.model';

@Component({

selector: 'app-employee-edit',

templateUrl: './employee-edit.component.html',

})

export class EmployeeEditComponent implements OnInit {

employee: Employee = { id: 0, name: '', department: '', salary: 0 };

constructor(

private route: ActivatedRoute,

private service: EmployeeService

) {}

ngOnInit(): void {

const id = Number(this.route.snapshot.paramMap.get('id'));

this.service.getEmployeeById(id).subscribe((emp) => {

this.employee = emp;

});

}

}

**employee-edit.component.html**

<h2>Edit Employee</h2>

<form>

<label>ID:</label>

<input type="text" [value]="employee.id" disabled /><br />

<label>Name:</label>

<input type="text" [(ngModel)]="employee.name" name="name" /><br />

<label>Department:</label>

<input type="text" [(ngModel)]="employee.department" name="dept" /><br />

<label>Salary:</label>

<input type="number" [(ngModel)]="employee.salary" name="salary" /><br />

</form>

**Create static employee list data using spring xml configuration**

**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

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

<!-- Reusable Skills -->

<bean id="skillJava" class="com.example.model.Skill">

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

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

</bean>

<bean id="skillAngular" class="com.example.model.Skill">

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

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

</bean>

<!-- Departments -->

<bean id="departmentHR" class="com.example.model.Department">

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

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

</bean>

<bean id="departmentIT" class="com.example.model.Department">

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

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

</bean>

<bean id="departmentFinance" class="com.example.model.Department">

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

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

</bean>

<!-- Employees -->

<bean id="employee1" class="com.example.model.Employee">

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

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

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

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

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

<property name="skills">

<list>

<ref bean="skillJava" />

</list>

</property>

</bean>

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

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

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

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

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

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

<property name="skills">

<list>

<ref bean="skillJava" />

<ref bean="skillAngular" />

</list>

</property>

</bean>

<bean id="employee3" class="com.example.model.Employee">

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

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

<property name="salary" value="72000" />

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

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

<property name="skills">

<list>

<ref bean="skillAngular" />

</list>

</property>

</bean>

<bean id="employee4" class="com.example.model.Employee">

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

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

<property name="salary" value="68000" />

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

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

<property name="skills">

<list>

<ref bean="skillJava" />

<ref bean="skillAngular" />

</list>

</property>

</bean>

<!-- Employee List -->

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

<constructor-arg>

<list>

<ref bean="employee1" />

<ref bean="employee2" />

<ref bean="employee3" />

<ref bean="employee4" />

</list>

</constructor-arg>

</bean>

</beans>

**EmployeeDao.java**

package com.example.dao;

import com.example.model.Employee;

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

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Repository;

import java.util.ArrayList;

import java.util.List;

@Repository

public class EmployeeDao {

private static List<Employee> EMPLOYEE\_LIST;

@Autowired

public EmployeeDao(ApplicationContext context) {

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

}

public List<Employee> getAllEmployees() {

return EMPLOYEE\_LIST;

}

}

**Application.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") // load the XML config

public class Application {

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

}

**Create REST service for department**

**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;

}

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; }

}

**employee.xml**

<!-- Department List -->

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

<constructor-arg>

<list>

<ref bean="departmentHR"/>

<ref bean="departmentIT"/>

<ref bean="departmentFinance"/>

</list>

</constructor-arg>

</bean>

**DepartmentDao.java**

package com.example.dao;

import com.example.model.Department;

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

import org.springframework.context.ApplicationContext;

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public class DepartmentDao {

private static List<Department> DEPARTMENT\_LIST;

@Autowired

public DepartmentDao(ApplicationContext context) {

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

}

public List<Department> getAllDepartments() {

return DEPARTMENT\_LIST;

}

}

**DepartmentService.java**

package com.example.service;

import com.example.dao.DepartmentDao;

import com.example.model.Department;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class DepartmentService {

@Autowired

private DepartmentDao departmentDao;

public List<Department> getAllDepartments() {

return departmentDao.getAllDepartments();

}

}

**DepartmentController.java**

package com.example.controller;

import com.example.model.Department;

import com.example.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("Fetching all departments...");

return departmentService.getAllDepartments();

}

}

**Application.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 Application {

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

}

**Output:**

