**ADDITIONAL HANDSON**

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

**EmployeeServiceApplication.java**

package com.example;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource; // ✅ You missed this

*@SpringBootApplication*

*@ImportResource*("classpath:employee.xml") // ✅ This loads XML bean definitions

public class EmployeeServiceApplication {

public static void main(String[] args) {

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

}

}

**AppConfig.java**

package com.example.config;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.ImportResource;

*@Configuration*

public class AppConfig {

}

**EmployeeControlle.java**

package com.example.controller;

import com.example.model.Employee;

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

import org.springframework.context.ApplicationContext;

import org.springframework.http.ResponseEntity;

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

import java.util.List;

*@RestController*

*@RequestMapping*("/api/employees")

*@CrossOrigin*(origins = "http://localhost:4200") // Allows Angular frontend to call this API

public class EmployeeController {

*@Autowired*

private ApplicationContext context; // Automatically loads beans from employee.xml

*@GetMapping*

public ResponseEntity<List<Employee>> getAllEmployees() {

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

return ResponseEntity.*ok*(employees); // returns 200 OK with list

}

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

public ResponseEntity<Employee> getEmployeeById(*@PathVariable* int id) {

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

return employees.stream()

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

.findFirst()

.map(ResponseEntity::*ok*) // returns 200 OK with employee

.orElse(ResponseEntity.*notFound*().build()); // returns 404 if not found

}

}

**HelloController.java**

package com.example.controller;

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

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

*@RestController*

public class HelloController {

*@GetMapping*("/")

public String home() {

return "Welcome to the Home Page!";

}

*@GetMapping*("/hello")

public String sayHello() {

return "Hello World!!";

}

}

**Employee.java**

package com.example.model;

public class Employee {

private int id;

private String name;

private String department;

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; }

}

**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">

<!-- Employee 1 -->

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

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

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

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

</bean>

<!-- Employee 2 -->

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

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

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

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

</bean>

<!-- List of Employees -->

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

<constructor-arg>

<list>

<ref bean="employee1"/>

<ref bean="employee2"/>

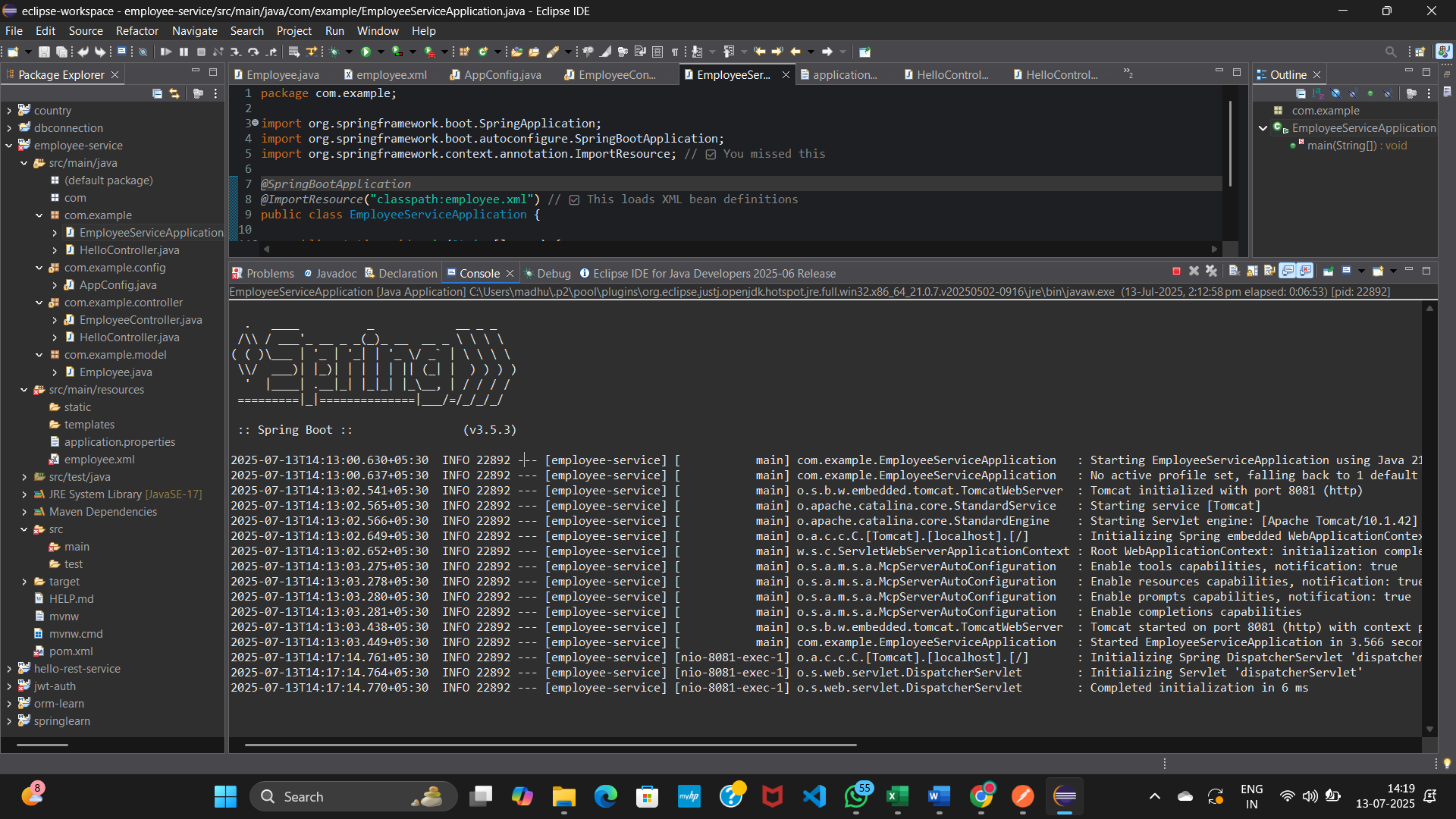
</list>

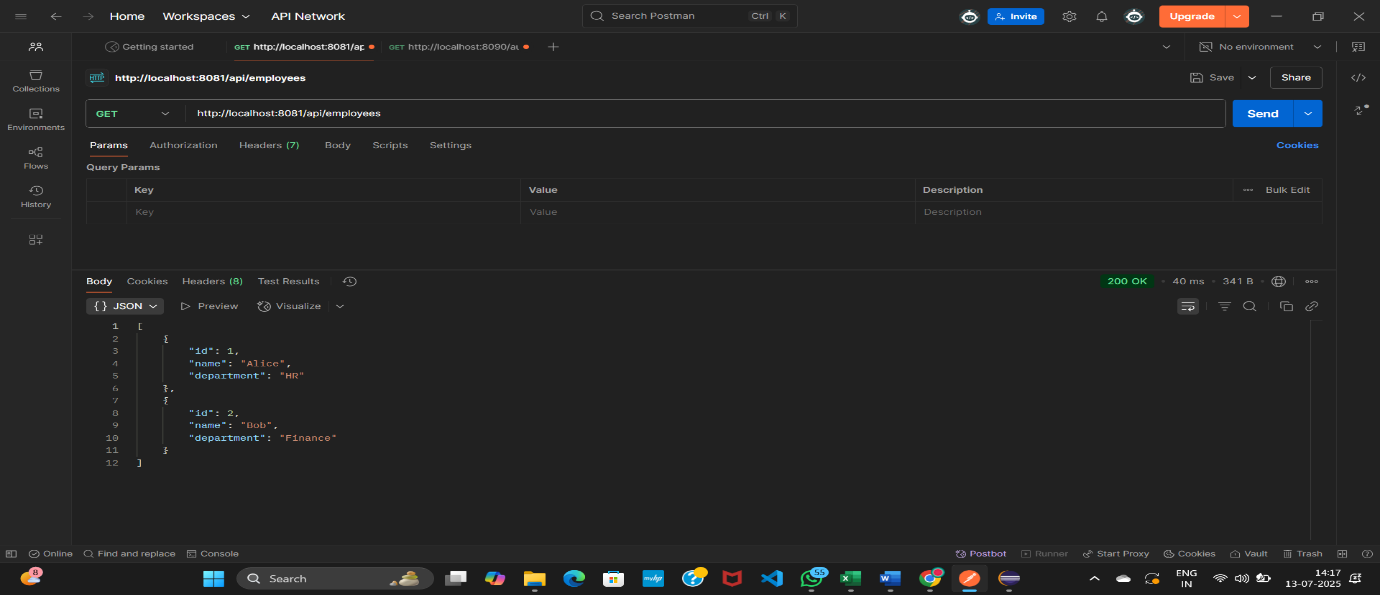
</constructor-arg>

</bean>

</beans>

**OUTPUT:**

****

****

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

**EmployeeServiceApplication.java**

package com.example;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource; // ✅ You missed this

*@SpringBootApplication*

*@ImportResource*("classpath:employee.xml") // ✅ This loads XML bean definitions

public class EmployeeServiceApplication {

public static void main(String[] args) {

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

}

}

**AppConfig.java**

package com.example.config;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.ImportResource;

*@Configuration*

public class AppConfig {

}

**EmployeeControlle.java**

package com.example.controller;

import com.example.model.Employee;

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

import org.springframework.context.ApplicationContext;

import org.springframework.http.ResponseEntity;

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

import java.util.List;

*@RestController*

*@RequestMapping*("/api/employees")

*@CrossOrigin*(origins = "http://localhost:4200") // Allows Angular frontend to call this API

public class EmployeeController {

*@Autowired*

private ApplicationContext context; // Automatically loads beans from employee.xml

*@GetMapping*

public ResponseEntity<List<Employee>> getAllEmployees() {

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

return ResponseEntity.*ok*(employees); // returns 200 OK with list

}

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

public ResponseEntity<Employee> getEmployeeById(*@PathVariable* int id) {

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

return employees.stream()

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

.findFirst()

.map(ResponseEntity::*ok*) // returns 200 OK with employee

.orElse(ResponseEntity.*notFound*().build()); // returns 404 if not found

}

}

**HelloController.java**

package com.example.controller;

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

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

*@RestController*

public class HelloController {

*@GetMapping*("/")

public String home() {

return "Welcome to the Home Page!";

}

*@GetMapping*("/hello")

public String sayHello() {

return "Hello World!!";

}

}

**Employee.java**

package com.example.model;

public class Employee {

private int id;

private String name;

private String department;

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; }

}

**Department.java**

package com.example.model;

public class Department {

private int id;

private String 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**

<?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">

<!-- Departments -->

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

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

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

</bean>

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

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

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

</bean>

<bean id="financeDept" 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="hrDept"/>

</bean>

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

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

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

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

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

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

</bean>

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

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

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

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

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

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

</bean>

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

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

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

<property name="salary" value="80000"/>

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

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

</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"/>

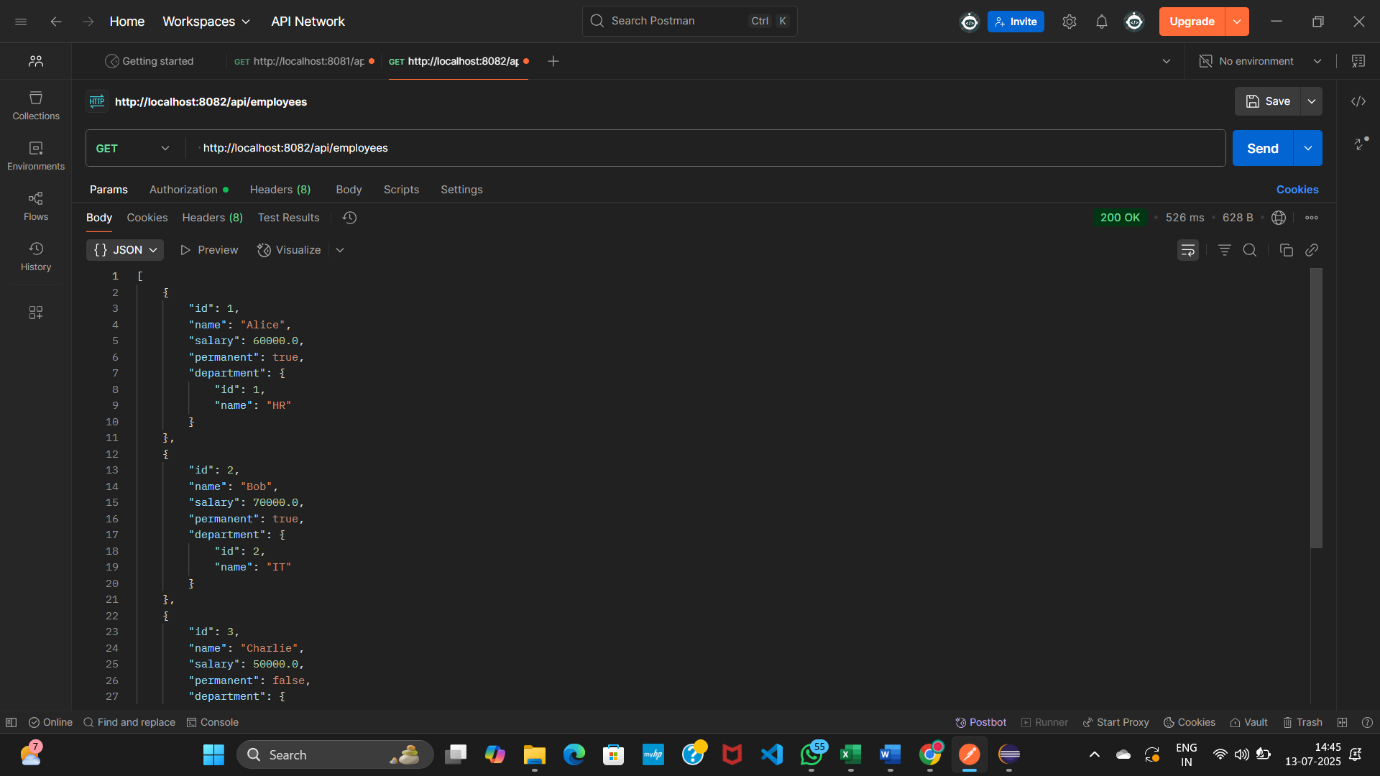
</list>

</constructor-arg>

</bean>

</beans>

**OUTPUT:**

****