

## Exercise 1: Problem Statement - Display Employee List and Edit Employee form using RESTful Web Service

### CODE:

#### Employee.java:

```
package com.example.employeeapi.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; }

}
```

#### employees.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="employeeList" class="java.util.ArrayList">

        <constructor-arg>

            <list>

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

                    <property name="id" value="101"/>

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

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

                </bean>

            </list>

        </constructor-arg>

    </bean>

</beans>
```

```

        <bean class="com.example.employeeapi.model.Employee">
            <property name="id" value="102"/>
            <property name="name" value="Valli"/>
            <property name="department" value="HR"/>
        </bean>

        <bean class="com.example.employeeapi.model.Employee">
            <property name="id" value="103"/>
            <property name="name" value="Ravi"/>
            <property name="department" value="Finance"/>
        </bean>

    </list>

</constructor-arg>

</bean>

</beans>

```

### **EmployeeService.java:**

```

package com.example.employeeapi.service;

import com.example.employeeapi.model.Employee;
import org.springframework.stereotype.Service;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import java.util.List;

@Service
public class EmployeeService {

    public List<Employee> getAllEmployees() {

        ClassPathXmlApplicationContext context = new
        ClassPathXmlApplicationContext("employees.xml");

        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.employeeapi.controller;

import com.example.employeeapi.model.Employee;
import com.example.employeeapi.service.EmployeeService;
import org.springframework.web.bind.annotation.*;
import org.springframework.beans.factory.annotation.Autowired;
import java.util.List;

@RestController
@RequestMapping("/api/employees")
@CrossOrigin(origins = "*") // Optional: allow access from frontend
public class EmployeeController {

    @Autowired
    private EmployeeService employeeService;

    @GetMapping
    public List<Employee> getAll() {
        return employeeService.getAllEmployees();
    }

    @GetMapping("/{id}")
    public Employee getById(@PathVariable int id) {
        return employeeService.getEmployeeById(id);
    }
}

```

## OUTPUT:

```
pretty-print :  
[{"id":101,"name":"Ram","department":"IT"}, {"id":102,"name":"Valli","department":"HR"}, {"id":103,"name":"Ravi","department":"Finance"}]
```

## Exercise 2: Create static employee list data using spring xml configuration

### CODE:

#### EmployeeController.java:

```
package com.example.employee_api.controller;

import com.example.employee_api.model.Employee;
import com.example.employee_api.service.EmployeeService;
import org.springframework.web.bind.annotation.*;
import org.springframework.beans.factory.annotation.Autowired;
import java.util.List;

@RestController
@RequestMapping("/api/employees")
@CrossOrigin(origins = "*") // Optional: allow access from frontend

public class EmployeeController {

    @Autowired
    private EmployeeService employeeService;

    @GetMapping
    public List<Employee> getAll() {
        return employeeService.getAllEmployees();
    }

    @GetMapping("/{id}")
    public Employee getById(@PathVariable int id) {
        return employeeService.getEmployeeById(id);
    }
}
```

#### EmployeeDao.java:

```
package com.example.employee_api.dao;

import com.example.employee_api.model.Employee;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;
```

```

import java.util.ArrayList;
import java.util.List;
@Repository
public class EmployeeDao {

    public static List<Employee> EMPLOYEE_LIST = new ArrayList<>();

    public EmployeeDao() {

        try {

            ClassPathXmlApplicationContext context =

                new ClassPathXmlApplicationContext("employee.xml");

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

            System.out.println(">>> Loaded employees from XML: " +
EMPLOYEE_LIST.size());

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

    public List<Employee> getAllEmployees() {

        return EMPLOYEE_LIST;

    }

}

```

### **Employee.java:**

```

package com.example.employee_api.model;

import java.util.List;

public class Employee {

    private int id;

    private String name;

    private String department;

    private List<String> skills;

```

```
public Employee() {}

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 List<String> getSkills() { return skills; }

public void setSkills(List<String> skills) { this.skills = skills; }

}
```

## OUTPUT:

```
pretty-print
[{"id":101,"name":"Ram","department":"IT","skills":["Java","Spring"]}, {"id":102,"name":"Valli","department":"HR","skills":["Communication","Recruitment"]}, {"id":103,"name":"Ravi","department":"Finance","skills":["Excel","Analytics"]}, {"id":104,"name":"Deepa","department":"IT","skills":["Java","Spring"]}]
```

### Exercise 3: Create REST service to gets all employees

#### CODE:

##### EmployeeController.java:

```
package com.example.employee_api.controller;

import com.example.employee_api.model.Employee;
import com.example.employee_api.service.EmployeeService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
@RequestMapping("/api")
public class EmployeeController {

    @Autowired
    private EmployeeService employeeService;

    @GetMapping("/employees")
    public List<Employee> getAllEmployees() {
        return employeeService.getAllEmployees();
    }
}
```

##### EmployeeDao.java:

```
package com.example.employee_api.dao;

import com.example.employee_api.model.Employee;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;
import java.util.ArrayList;
import java.util.List;

@Repository
public class EmployeeDao {

    public static List<Employee> EMPLOYEE_LIST = new ArrayList<>();
}
```



```

public static void main(String[] args) {
    try {
        ClassPathXmlApplicationContext context =
            new ClassPathXmlApplicationContext("employee.xml");
        List<Employee> list = (List<Employee>) context.getBean("employeeList");
        System.out.println("TEST: Loaded " + list.size() + " employees");
        for (Employee emp : list) {
            System.out.println(emp.getId() + " - " + emp.getName());
        }
    } catch (Exception e) {
        System.out.println("Failed to load employee.xml");
        e.printStackTrace();
    }
}

public List<Employee> getAllEmployees() {
    return EMPLOYEE_LIST;
}
}

```

### **Employee.java:**

```

package com.example.employee_api.model;
import java.util.List;
public class Employee {
    private int id;
    private String name;
    private String department;
    private List<String> skills;
    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 List<String> getSkills() { return skills; }

public void setSkills(List<String> skills) { this.skills = skills; }

}
```

### **EmployeeService.java:**

```
package com.example.employee_api.service;

import com.example.employee_api.dao.EmployeeDao;
import com.example.employee_api.model.Employee;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import java.util.List;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeDao employeeDao;

    @Transactional
    public List<Employee> getAllEmployees() {
        return employeeDao.getAllEmployees();
    }

}
```

# OUTPUT:

Collections

Environments

Flows

History

http://localhost:8080/api/employees

GET

http://localhost:8080/api/employees

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Query Params

Key	Value	Description
Key	Value	Description

Body

Cookies

Headers (5)

Test Results

JSON

Preview

Visualize

200 OK

4 ms

166 B

1

## Exercise 4: Create REST service for department

### CODE:

#### DepartmentController.java:

```
package com.example.employee_api.controller;

import com.example.employee_api.service.DepartmentService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
@RequestMapping("/api")
public class DepartmentController {

    @Autowired
    private DepartmentService departmentService;

    @GetMapping("/departments")
    public List<String> getAllDepartments() {
        System.out.println("/departments endpoint called");
        return departmentService.getAllDepartments();
    }
}
```

#### DepartmentDao.java:

```
package com.example.employee_api.dao;

import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;
import java.util.ArrayList;
import java.util.List;

@Repository
public class DepartmentDao {

    private static List<String> DEPARTMENT_LIST = new ArrayList<>();

    public DepartmentDao() {
```

```

    try {
        ClassPathXmlApplicationContext context =
            new ClassPathXmlApplicationContext("employee.xml")

        DEPARTMENT_LIST = (List<String>) context.getBean("departmentList");

        System.out.println("Loaded Departments: " + DEPARTMENT_LIST);
    } catch (Exception e) {
        System.out.println("Failed to load departmentList from XML");
        e.printStackTrace();
    }
}

public List<String> getAllDepartments() {
    return DEPARTMENT_LIST;
}
}

```

### **DepartmentService.java:**

```

package com.example.employee_api.service;

import com.example.employee_api.dao.DepartmentDao;
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<String> getAllDepartments() {
        return departmentDao.getAllDepartments();
    }
}

```

OUTPUT:

Collections

Environments

Flows

History

http://localhost:8080/api/departments

GET

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body

Cookies

Headers (5)

Test Results

200 OK

206 ms

166 B

JSON

Preview

Visualize

1