# **Create Static Employee List Data Using Spring XML Configuration**

## **Step 1: Create employee.xml for Static Data Configuration**

**src/main/resources/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">

<!-- Department 1 -->

<bean id="department1" class="com.cognizant.springlearn.Department">

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

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

</bean>

<!-- Department 2 -->

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

<property name="id" value="102"/>

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

</bean>

<!-- Department 3 -->

<bean id="department3" class="com.cognizant.springlearn.Department">

<property name="id" value="103"/>

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

</bean>

<!-- Reuse existing skills -->

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

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

</bean>

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

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

</bean>

<bean id="skill3" class="com.cognizant.springlearn.Skill">

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

</bean>

<bean id="skill4" class="com.cognizant.springlearn.Skill">

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

</bean>

<!-- Employee 1 -->

<bean id="employee1" class="com.cognizant.springlearn.Employee">

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

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

<property name="department" ref="department1"/>

<property name="skills">

<list>

<ref bean="skill1"/>

<ref bean="skill2"/>

</list>

</property>

</bean>

<!-- Employee 2 -->

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

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

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

<property name="department" ref="department2"/>

<property name="skills">

<list>

<ref bean="skill3"/>

<ref bean="skill4"/>

</list>

</property>

</bean>

<!-- Employee 3 -->

<bean id="employee3" class="com.cognizant.springlearn.Employee">

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

<property name="name" value="Mary Johnson"/>

<property name="department" ref="department3"/>

<property name="skills">

<list>

<ref bean="skill1"/>

<ref bean="skill3"/>

</list>

</property>

</bean>

<!-- Employee 4 -->

<bean id="employee4" class="com.cognizant.springlearn.Employee">

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

<property name="name" value="James Williams"/>

<property name="department" ref="department2"/>

<property name="skills">

<list>

<ref bean="skill2"/>

<ref bean="skill4"/>

</list>

</property>

</bean>

<!-- Employee list -->

<bean id="employeeDao" class="com.cognizant.springlearn.dao.EmployeeDao">

<property name="employeeList">

<list>

<ref bean="employee1"/>

<ref bean="employee2"/>

<ref bean="employee3"/>

<ref bean="employee4"/>

</list>

</property>

</bean>

</beans>

## **Step 2: Create Employee, Department, and Skill Classes**

### **src/main/java/com/cognizant/springlearn/Employee.java**

package com.cognizant.springlearn;

import java.util.List;

public class Employee {

private int id;

private String name;

private Department department;

private List<Skill> skills;

public Employee() {}

// Getters and setters

@Override

public String toString() {

return "Employee{id=" + id + ", name='" + name + "', department=" + department + ", skills=" + skills + "}";

}

}

### **src/main/java/com/cognizant/springlearn/Department.java**

package com.cognizant.springlearn;

public class Department {

private int id;

private String name;

public Department() {}

// Getters and setters

@Override

public String toString() {

return "Department{id=" + id + ", name='" + name + "'}";

}

}

### **src/main/java/com/cognizant/springlearn/Skill.java**

package com.cognizant.springlearn;

public class Skill {

private String name;

public Skill() {}

// Getters and setters

@Override

public String toString() {

return "Skill{name='" + name + "'}";

}

}

## **Step 3: Create EmployeeDao to Manage the Employee List**

**src/main/java/com/cognizant/springlearn/dao/EmployeeDao.java**

package com.cognizant.springlearn.dao;

import com.cognizant.springlearn.Employee;

import java.util.ArrayList;

import java.util.List;

public class EmployeeDao {

private static List<Employee> EMPLOYEE\_LIST;

public EmployeeDao(List<Employee> employeeList) {

EMPLOYEE\_LIST = employeeList;

}

public List<Employee> getAllEmployees() {

return EMPLOYEE\_LIST;

}

}

## **Step 4: Testing and Validation**

package com.cognizant.springlearn;

import com.cognizant.springlearn.dao.EmployeeDao;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

public class Main {

public static void main(String[] args) {

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

EmployeeDao employeeDao = context.getBean(EmployeeDao.class);

List<Employee> employees = employeeDao.getAllEmployees();

employees.forEach(System.out::println);

}

}

### **Output:**

Employee{id=1, name='John Doe', department=Department{id=101, name='HR'}, skills=[Skill{name='Java'}, Skill{name='SQL'}]}

Employee{id=2, name='Jane Smith', department=Department{id=102, name='IT'}, skills=[Skill{name='Python'}, Skill{name='AWS'}]}

Employee{id=3, name='Mary Johnson', department=Department{id=103, name='Finance'}, skills=[Skill{name='Java'}, Skill{name='Python'}]}

Employee{id=4, name='James Williams', department=Department{id=102, name='IT'}, skills=[Skill{name='SQL'}, Skill{name='AWS'}]}

# **Create REST Service to gets All Employees**

### **1. EmployeeService.java**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.Employee;

import com.cognizant.springlearn.dao.EmployeeDao;

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();

}

}

### **2. EmployeeController.java**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.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();

}

}

### **3. Enable Component Scanning & Load XML**

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ImportResource;

@SpringBootApplication

@ImportResource("classpath:employee.xml") // To load employee beans

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

### **4. Ensure XML Configuration is Correct**

<bean id="employeeDao" class="com.cognizant.springlearn.dao.EmployeeDao">

<constructor-arg>

<list>

<ref bean="employee1"/>

<ref bean="employee2"/>

<ref bean="employee3"/>

<ref bean="employee4"/>

</list>

</constructor-arg>

</bean>

### **5. Testing with Postman**

#### **URL:**

http://localhost:8080/employees

#### **Method:**

GET

#### **Response (JSON):**

[

{

"id": 1,

"name": "John Doe",

"department": {

"id": 101,

"name": "HR"

},

"skills": [

{"name": "Java"},

{"name": "SQL"}

]

},

{

"id": 2,

"name": "Jane Smith",

"department": {

"id": 102,

"name": "IT"

},

"skills": [

{"name": "Python"},

{"name": "AWS"}

]

},

...

]

# **Create REST Service for Department List**

### **1. department.xml Configuration**

src/main/resources/department.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">

<!-- Department Beans -->

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

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

<property name="name" value="Human Resources" />

</bean>

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

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

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

</bean>

<bean id="dept3" class="com.cognizant.springlearn.Department">

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

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

</bean>

<!-- DAO Bean with Department List -->

<bean id="departmentDao" class="com.cognizant.springlearn.dao.DepartmentDao">

<constructor-arg>

<list>

<ref bean="dept1"/>

<ref bean="dept2"/>

<ref bean="dept3"/>

</list>

</constructor-arg>

</bean>

</beans>

### **2. DepartmentDao.java**

package com.cognizant.springlearn.dao;

import com.cognizant.springlearn.Department;

import java.util.List;

public class DepartmentDao {

private static List<Department> DEPARTMENT\_LIST;

public DepartmentDao(List<Department> departmentList) {

DEPARTMENT\_LIST = departmentList;

}

public List<Department> getAllDepartments() {

return DEPARTMENT\_LIST;

}

}

### **3. DepartmentService.java**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.Department;

import com.cognizant.springlearn.dao.DepartmentDao;

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 DepartmentService {

@Autowired

private DepartmentDao departmentDao;

@Transactional

public List<Department> getAllDepartments() {

return departmentDao.getAllDepartments();

}

}

### **4. DepartmentController.java**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.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.debug("START: getAllDepartments()");

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

LOGGER.debug("END: getAllDepartments()");

return departments;

}

}

### **5. Department.java**

package com.cognizant.springlearn;

public class Department {

private int id;

private String name;

public Department() {}

// Getters and Setters

@Override

public String toString() {

return "Department{id=" + id + ", name='" + name + "'}";

}

}

### **6. SpringLearnApplication.java**

@SpringBootApplication

@ImportResource({"classpath:employee.xml", "classpath:department.xml"})

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

### **7. Testing via Postman**

#### **URL:**

http://localhost:8080/departments

#### **Method:**

GET

#### **Response:**

[

{

"id": 1,

"name": "Human Resources"

},

{

"id": 2,

"name": "Finance"

},

{

"id": 3,

"name": "IT"

}

]