**Exercise 3: Employee Management System - Creating Repositories**

**Business Scenario:**

Create repositories for Employee and Department entities to perform CRUD operations.

**Instructions:**

1. **Overview of Spring Data Repositories:**
   * Learn the benefits of using Spring Data repositories.
2. **Creating Repositories:**
   * Create **EmployeeRepository** and **DepartmentRepository** interfaces extending **JpaRepository**.
   * Define derived query methods in these repositories.

**Solution**

**Employee.java**

package com.employee.employeemanagement.entity;

import jakarta.persistence.\*;

*@Entity*

*@Table*(name = "employees")

public class Employee {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

private Long id;

private String name;

private String email;

*@ManyToOne*

*@JoinColumn*(name = "department\_id")

private Department department;

// Constructors, Getters, Setters, etc.

public Employee() {

}

public Employee(String name, String email, Department department) {

this.name = name;

this.email = email;

this.department = department;

}

*@Override*

public String toString() {

return "Employee{" +

"id=" + id +

", name='" + name + '\'' +

", email='" + email + '\'' +

", department=" + department.getName() +

'}';

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public Department getDepartment() {

return department;

}

public void setDepartment(Department department) {

this.department = department;

}

}

**Department.java**

package com.employee.employeemanagement.entity;

import jakarta.persistence.\*;

*@Entity*

*@Table*(name = "departments")

public class Department {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

private Long id;

private String name;

// Constructors, Getters, Setters

public Department() {}

public Department(String name) {

this.name = name;

}

*@Override*

public String toString() {

return "Department{" +

"id=" + id +

", name='" + name + '\'' +

'}';

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**EmployeeRepository.java**

package com.employee.employeemanagement.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.employee.employeemanagement.entity.Employee;

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Derived query methods

List<Employee> findByName(String name);

List<Employee> findByEmail(String email);

List<Employee> findByDepartmentName(String departmentName);

}

**DepartmentRepository.java**

package com.employee.employeemanagement.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.employee.employeemanagement.entity.Department;

import java.util.List;

public interface DepartmentRepository extends JpaRepository<Department, Long> {

// Derived query methods

List<Department> findByName(String name);

}

**EmployeeManagementSystemApplication.java**

package com.employee.employeemanagement;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.employee.employeemanagement.entity.Department;

import com.employee.employeemanagement.entity.Employee;

import com.employee.employeemanagement.repository.DepartmentRepository;

import com.employee.employeemanagement.repository.EmployeeRepository;

@SpringBootApplication(scanBasePackages = "com.employee.employeemanagement")

public class EmployeeManagementSystemApplication implements CommandLineRunner {

@Autowired

private DepartmentRepository departmentRepository;

@Autowired

private EmployeeRepository employeeRepository;

public static void main(String[] args) {

SpringApplication.run(EmployeeManagementSystemApplication.class, args);

}

@Override

public void run(String... args) throws Exception {

System.out.println();

System.out.println("Employee Management System is running successfully!");

// Create and save a Department instance

Department cseDepartment = new Department("CSE");

departmentRepository.save(cseDepartment);

// Create and save Employee instances

Employee employee1 = new Employee("Shobika", "shobika@example.com", cseDepartment);

Employee employee2 = new Employee("Monika", "monika@example.com", cseDepartment);

employeeRepository.save(employee1);

employeeRepository.save(employee2);

// Print employee details to the console

System.out.println("Created Employee 1: " + employee1.getName() + ", Email: " + employee1.getEmail() + ", Department: " + employee1.getDepartment().getName());

System.out.println("Created Employee 2: " + employee2.getName() + ", Email: " + employee2.getEmail() + ", Department: " + employee2.getDepartment().getName());

// Test derived query methods

System.out.println("Find Employees by Name: " + employeeRepository.findByName("Shobika"));

System.out.println("Find Employees by Name: " + employeeRepository.findByName("Monika"));

System.out.println("Find Departments by Name: " + departmentRepository.findByName("CSE"));

}

}

