Step 1: Define Interface-Based Projections

Create interface-based projections to fetch specific data subsets from the Employee entity.

EmployeeSummary.java:

package com.example.employeemanagementsystem.projection;

public interface EmployeeSummary {

Long getId();

String getName();

String getEmail();

}

EmployeeDetails.java:

package com.example.employeemanagementsystem.projection;

public interface EmployeeDetails {

Long getId();

String getName();

String getEmail();

String getJobTitle();

DepartmentSummary getDepartment();

}

interface DepartmentSummary {

Long getId();

String getName();

}

Step 2: Define Class-Based Projections

Create class-based projections to fetch specific data subsets from the Department entity.

DepartmentOverview.java:

package com.example.employeemanagementsystem.projection;

public class DepartmentOverview {

private Long id;

private String name;

private int employeeCount;

public DepartmentOverview(Long id, String name, int employeeCount) {

this.id = id;

this.name = name;

this.employeeCount = employeeCount;

}

// getters and setters

}

Step 3: Use @Value and Constructor Expressions

Use @Value and constructor expressions to control the fetched data in the projections.

EmployeeRepository.java:

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.entity.Employee;

import com.example.employeemanagementsystem.projection.EmployeeDetails;

import com.example.employeemanagementsystem.projection.EmployeeSummary;

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

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

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

@Query("SELECT e.id AS id, e.name AS name, e.email AS email FROM Employee e")

List<EmployeeSummary> findEmployeeSummaries();

@Query("SELECT new com.example.employeemanagementsystem.projection.EmployeeDetails(e.id, e.name, e.email, e.jobTitle, d) FROM Employee e JOIN e.department d")

List<EmployeeDetails> findEmployeeDetails();

}

DepartmentRepository.java:

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.entity.Department;

import com.example.employeemanagementsystem.projection.DepartmentOverview;

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

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

public interface DepartmentRepository extends JpaRepository<Department, Long> {

@Query("SELECT new com.example.employeemanagementsystem.projection.DepartmentOverview(d.id, d.name, SIZE(d.employees)) FROM Department d")

List<DepartmentOverview> findDepartmentOverviews();

}