package com.example.employeemanagementsystem.projection;

public interface EmployeeNameProjection {

String getName();

String getEmail();

}

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.Employee;

import com.example.employeemanagementsystem.projection.EmployeeNameProjection;

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

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Fetch only name and email fields

List<EmployeeNameProjection> findAllProjectedBy();

}

package com.example.employeemanagementsystem.dto;

public class EmployeeDTO {

private String name;

private String departmentName;

public EmployeeDTO(String name, String departmentName) {

this.name = name;

this.departmentName = departmentName;

}

// Getters and Setters

}

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.Employee;

import com.example.employeemanagementsystem.dto.EmployeeDTO;

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

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

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Fetch name and department name using class-based projection

@Query("SELECT new com.example.employeemanagementsystem.dto.EmployeeDTO(e.name, d.name) " +

"FROM Employee e JOIN e.department d")

List<EmployeeDTO> findAllEmployeeDTOs();

}

package com.example.employeemanagementsystem.projection;

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

public interface EmployeeDetailProjection {

String getName();

@Value("#{target.name + ' (' + target.department.name + ')'}")

String getEmployeeWithDepartment();

}

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.Employee;

import com.example.employeemanagementsystem.projection.EmployeeDetailProjection;

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

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Fetch employee name and formatted department name

List<EmployeeDetailProjection> findAllEmployeeDetailsProjectedBy();

}

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.Employee;

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

import org.springframework.data.repository.query.Param;

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Dynamic projections based on the passed type

<T> List<T> findByDepartmentName(@Param("departmentName") String departmentName, Class<T> type);

}

package com.example.employeemanagementsystem.service;

import com.example.employeemanagementsystem.dto.EmployeeDTO;

import com.example.employeemanagementsystem.projection.EmployeeNameProjection;

import com.example.employeemanagementsystem.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

public List<EmployeeNameProjection> getEmployeeNamesByDepartment(String departmentName) {

return employeeRepository.findByDepartmentName(departmentName, EmployeeNameProjection.class);

}

public List<EmployeeDTO> getEmployeeDTOsByDepartment(String departmentName) {

return employeeRepository.findByDepartmentName(departmentName, EmployeeDTO.class);

}

}