**SpringDataJPAHibernate and Emplomement Management System**

**Mvn/wrapper**

**Maven-wrapper.properties**

# Licensed to the Apache Software Foundation (ASF) under one

# or more contributor license agreements. See the NOTICE file

# distributed with this work for additional information

# regarding copyright ownership. The ASF licenses this file

# to you under the Apache License, Version 2.0 (the

# "License"); you may not use this file except in compliance

# with the License. You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing,

# software distributed under the License is distributed on an

# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

# KIND, either express or implied. See the License for the

# specific language governing permissions and limitations

# under the License.

wrapperVersion=3.3.2

distributionType=only-script

distributionUrl=https://repo.maven.apache.org/maven2/org/apache/maven/apache-maven/3.9.10/apache-maven-3.9.10-bin.zip

**/test/java/com/example**

**EmploymentManagamentSystemApplicationTests.java**

package com.example.Employee\_Management\_System;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class EmployeeManagementSystemApplicationTests {

@Test

void contextLoads() {

}

}

/main/resources

Application.properties

spring.application.name=Employee-Management-System

spring.datasource.url=${DB\_URL:jdbc:h2:mem:testdb}

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=${DB\_USER:sa}

spring.datasource.password=${DB\_PASS:password}

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.h2.console.enabled=true

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.format\_sql=true

spring.jpa.properties.hibernate.order\_inserts=true

spring.jpa.properties.hibernate.order\_updates=true

spring.jpa.properties.hibernate.jdbc.batch\_size=30

logging.level.org.springframework.web=DEBUG

**/main/java/com/example**

**EmplomentManagementSystemApplication.java**

package com.example.Employee\_Management\_System;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.data.jpa.repository.config.EnableJpaAuditing;

@SpringBootApplication

@EnableJpaAuditing

public class EmployeeManagementSystemApplication {

public static void main(String[] args) {

SpringApplication.run(EmployeeManagementSystemApplication.class, args);

}

}

**example/audit/auitable.java**

package com.example.Employee\_Management\_System.audit;

import jakarta.persistence.\*;

import lombok.Getter;

import lombok.Setter;

import org.springframework.data.annotation.\*;

import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import java.time.LocalDateTime;

@MappedSuperclass

@EntityListeners(AuditingEntityListener.class)

@Getter

@Setter

public abstract class Auditable {

@CreatedDate

@Column(updatable = false)

private LocalDateTime createdDate;

@LastModifiedDate

private LocalDateTime lastModifiedDate;

@CreatedBy

private String createdBy;

@LastModifiedBy

private String modifiedBy;

}

**Example/controller/EmployeeController.java**

package com.example.Employee\_Management\_System.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.\*;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.PageRequest;

import org.springframework.data.domain.Sort;

import org.springframework.http.ResponseEntity;

import com.example.Employee\_Management\_System.entity.Employee;

import com.example.Employee\_Management\_System.projection.EmployeeNameEmailView;

import com.example.Employee\_Management\_System.service.EmployeeService;

import com.example.Employee\_Management\_System.repository.EmployeeRepository;

import java.util.List;

@RestController

@RequestMapping("/api/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@Autowired

private EmployeeRepository employeeRepository;

@GetMapping

public List<Employee> getAllEmployees() {

return employeeRepository.findAll();

}

@GetMapping("/page")

public Page<Employee> getEmployeesPage(

@RequestParam(defaultValue = "0") int page,

@RequestParam(defaultValue = "5") int size,

@RequestParam(defaultValue = "name") String sortBy) {

return employeeRepository.findAll(PageRequest.of(page, size, Sort.by(sortBy)));

}

@GetMapping("/{id}")

public Employee getEmployeeById(@PathVariable Long id) {

return employeeRepository.findById(id).orElse(null);

}

@GetMapping("/projection")

public List<EmployeeNameEmailView> getEmployeeNameEmail() {

return employeeRepository.findAllBy();

}

@PostMapping

public Employee createEmployee(@RequestBody Employee employee) {

return employeeRepository.save(employee);

}

@PostMapping("/employees/batch")

public ResponseEntity<?> saveEmployees(@RequestBody List<Employee> employees) {

employeeService.saveAllEmployees(employees);

return ResponseEntity.ok("Batch inserted successfully!");

}

@PostMapping("/batch")

public String saveAllEmployees(@RequestBody List<Employee> employees) {

employeeService.saveAllEmployees(employees);

return "Batch save successful!";

}

@PutMapping("/{id}")

public Employee updateEmployee(@PathVariable Long id, @RequestBody Employee updated) {

return employeeRepository.findById(id).map(e -> {

e.setName(updated.getName());

e.setEmail(updated.getEmail());

e.setDepartment(updated.getDepartment());

return employeeRepository.save(e);

}).orElse(null);

}

@DeleteMapping("/{id}")

public void deleteEmployee(@PathVariable Long id) {

employeeRepository.deleteById(id);

}

}

**Controller/DepartmentController.java**

package com.example.Employee\_Management\_System.controller;

import com.example.Employee\_Management\_System.entity.Department;

import com.example.Employee\_Management\_System.repository.DepartmentRepository;

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

import org.springframework.web.bind.annotation.\*;

import org.springframework.data.domain.Page;

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

import java.util.List;

@RestController

@RequestMapping("/api/departments")

public class DepartmentController {

@Autowired

private DepartmentRepository departmentRepository;

@GetMapping

public List<Department> getAllDepartments() {

return departmentRepository.findAll();

}

@GetMapping("/{id}")

public Department getDepartmentById(@PathVariable Long id) {

return departmentRepository.findById(id).orElse(null);

}

@PostMapping

public Department createDepartment(@RequestBody Department department) {

return departmentRepository.save(department);

}

@PutMapping("/{id}")

public Department updateDepartment(@PathVariable Long id, @RequestBody Department updated) {

return departmentRepository.findById(id).map(d -> {

d.setName(updated.getName());

return departmentRepository.save(d);

}).orElse(null);

}

@DeleteMapping("/{id}")

public void deleteDepartment(@PathVariable Long id) {

departmentRepository.deleteById(id);

}

}

**/example/entity/Employee.java**

package com.example.Employee\_Management\_System.entity;

import com.fasterxml.jackson.annotation.JsonManagedReference;

import com.fasterxml.jackson.annotation.JsonBackReference;

import com.example.Employee\_Management\_System.audit.Auditable;

import jakarta.persistence.\*;

import lombok.\*;

@NamedQuery(name = "Employee.findByDepartmentName",

query = "SELECT e FROM Employee e WHERE e.department.name = ?1")

@Entity

@Table(name = "employees")

@Data

@NoArgsConstructor

@AllArgsConstructor

@org.hibernate.annotations.DynamicUpdate

public class Employee extends Auditable{

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

@ManyToOne

@JoinColumn(name = "department\_id")

@JsonBackReference

private Department department;

}

**/entity/Department.java**

package com.example.Employee\_Management\_System.entity;

import com.fasterxml.jackson.annotation.JsonManagedReference;

import com.fasterxml.jackson.annotation.JsonBackReference;

import com.example.Employee\_Management\_System.audit.Auditable;

import jakarta.persistence.\*;

import lombok.\*;

import java.util.List;

@Entity

@Table(name = "departments")

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Department extends Auditable{

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

@OneToMany(mappedBy = "department", cascade = CascadeType.ALL)

@JsonManagedReference

private List<Employee> employees;

}

Example/projection/EmployeeNameEmailView.java

package com.example.Employee\_Management\_System.projection;

public interface EmployeeNameEmailView {

String getName();

String getEmail();

}

**/example/repository/EmployeeRepository.java**

package com.example.Employee\_Management\_System.repository;

import com.example.Employee\_Management\_System.entity.Employee;

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

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

import org.springframework.stereotype.Repository;

import java.util.List;

import org.springframework.data.domain.Page;

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

import com.example.Employee\_Management\_System.projection.EmployeeNameEmailView;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Derived query method

List<Employee> findByNameContainingIgnoreCase(String name);

List<EmployeeNameEmailView> findAllBy();

// Custom JPQL query

@Query("SELECT e FROM Employee e WHERE e.email = :email")

Employee findByEmail(@Param("email") String email);

// Named Query (see entity below)

List<Employee> findByDepartmentName(String name);

}

**/repository/DepartmentRepository.java**

package com.example.Employee\_Management\_System.repository;

import com.example.Employee\_Management\_System.entity.Department;

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

import org.springframework.stereotype.Repository;

@Repository

public interface DepartmentRepository extends JpaRepository<Department, Long> {

}

**/example/service/EmployeeService.java**

package com.example.Employee\_Management\_System.service;

import com.example.Employee\_Management\_System.entity.Employee;

import com.example.Employee\_Management\_System.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;

// Batch save method (Hibernate batch insert support)

public void saveAllEmployees(List<Employee> employees) {

employeeRepository.saveAll(employees);

}

// Optional: individual operations can also go here later

}