**4.SPRINGDATAJPA-Employee Management System**

**Project structure:**

EmployeeManagementSystem

│

├── pom.xml

├── src

│ └── main

│ ├── java

│ │ └── com

│ │ └── example

│ │ └── ems

│ │ ├── EmployeeManagementSystemApplication.java

│ │

│ │ ├── config

│ │ │ └── JpaAuditingConfig.java

│ │

│ │ ├── controller

│ │ │ ├── EmployeeController.java

│ │ │ └── DepartmentController.java

│ │

│ │ ├── entity

│ │ │ ├── Employee.java

│ │ │ └── Department.java

│ │

│ │ ├── projection

│ │ │ ├── EmployeeNameOnly.java

│ │ │ └── DepartmentSummary.java

│ │

│ │ ├── repository

│ │ │ ├── EmployeeRepository.java

│ │ │ └── DepartmentRepository.java

│ │

│ │ └── service

│ │ └── EmployeeService.java (optional)

│

│ └── resources

│ ├── application.properties

│ ├── static

│ └── templates

│

└─── target (generated)

**EmployeeController.java**

package com.example.ems.controller;

import com.example.ems.entity.Employee;

import com.example.ems.repository.EmployeeRepository;

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

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

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

import java.util.List;

@RestController

@RequestMapping("/employees")

public class EmployeeController {

    @Autowired

    private EmployeeRepository repo;

    @PostMapping

    public Employee create(@RequestBody Employee emp) {

        return repo.save(emp);

    }

    @GetMapping

    public List<Employee> all() {

        return repo.findAll();

    }

    @GetMapping("/paged")

    public Page<Employee> getPagedEmployees(@RequestParam int page, @RequestParam int size) {

        return repo.findAll(PageRequest.of(page, size));

    }

    @GetMapping("/sorted")

    public List<Employee> getSortedEmployees(@RequestParam String sortBy) {

        return repo.findAll(Sort.by(sortBy));

    }

    @GetMapping("/paged-sorted")

    public Page<Employee> getPagedSorted(@RequestParam int page, @RequestParam int size, @RequestParam String sortBy) {

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

    }

}

**Department.java**

package com.example.ems.entity;

import jakarta.persistence.\*;

import java.util.List;

@Entity

public class Department {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    private String name;

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

    private List<Employee> employees;

    // Getters and Setters

}

**Employee.java**

package com.example.ems.entity;

import jakarta.persistence.\*;

import org.springframework.data.annotation.CreatedDate;

import org.springframework.data.annotation.LastModifiedDate;

import java.time.LocalDateTime;

@Entity

@NamedQuery(name = "Employee.findByEmail", query = "SELECT e FROM Employee e WHERE e.email = ?1")

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;

    @CreatedDate

    private LocalDateTime createdDate;

    @LastModifiedDate

    private LocalDateTime lastModifiedDate;

    // Getters and Setters

}

**EmployeeNameOnly.java**

package com.example.ems.projection;

public interface EmployeeNameOnly {

    String getName();

}

**EmployeeManagementSystemApplication.java**

package com.example.ems;

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);

    }

}

**Application.properties**

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=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.jdbc.batch\_size=30

**Pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

         https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>

    <artifactId>EmployeeManagementSystem</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <packaging>jar</packaging>

    <name>EmployeeManagementSystem</name>

    <description>Spring Boot Employee Management System</description>

    <!-- ✅ Use Spring Boot 3.x for jakarta.persistence support -->

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.1.4</version>

        <relativePath/>

    </parent>

    <properties>

        <java.version>17</java.version>

    </properties>

    <dependencies>

        <!-- Spring Data JPA -->

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-data-jpa</artifactId>

        </dependency>

        <!-- Spring Web -->

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <!-- H2 In-Memory Database -->

         <dependency>

            <groupId>com.h2database</groupId>

            <artifactId>h2</artifactId>

            <scope>runtime</scope>

        </dependency>

        <!-- Lombok -->

        <dependency>

            <groupId>org.projectlombok</groupId>

            <artifactId>lombok</artifactId>

            <optional>true</optional>

        </dependency>

        <!-- Enable JPA Auditing -->

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-aop</artifactId>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <!-- Spring Boot Plugin -->

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>

**OUTPUT:**



