**Exp-2 Pagination and Sorting in Spring Data JPA**

**1.pom.xml Dependencies**

<?xml version="1.0" encoding="UTF-8"?>

<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>

<parent>

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

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

<version>3.5.4</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com</groupId>

<artifactId>PaginationandSortingApplication</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Book</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>21</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<dependency>

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**2. application.properties**

spring.application.name=Book

spring.datasource.url=jdbc:mysql://localhost:3306/new

spring.datasource.username=root

spring.datasource.password=naga@1331

spring.jpa.hibernate.ddl-auto=create-drop

spring.jpa.show-sql=true

server.port=8843

**3. Book.java Class**

package com.example.demo;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.*IDENTITY*)

private long id;

private String title;

private String author;

public Book() {}

public Book(String title, String author) {

this.title = title;

this.author = author;

}

@Override

public String toString() {

return "Book{id=" + id + ", title='" + title + "', author='" + author + "'}";

}

// getters and setters

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getTitle() { return title; }

public void setTitle(String title) { this.title = title; }

public String getAuthor() { return author; }

public void setAuthor(String author) { this.author = author; }

}

**4. BookRepository Interface**

package com.example.demo;

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

public interface BookRepository extends JpaRepository<Book, Long>{

}

**5. BookController.java**

package com.example.demo;

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

import org.springframework.data.domain.Page;

import org.springframework.data.domain.PageRequest;

import org.springframework.data.domain.Pageable;

import org.springframework.data.domain.Sort;

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

@RestController

@RequestMapping("/books")

public class BookController {

@Autowired

private BookRepository bookRepository;

// Insert sample data if empty

@GetMapping("/init")

public String initData() {

if (bookRepository.count() == 0) {

bookRepository.save(new Book("Spring Boot Basics", "John"));

bookRepository.save(new Book("Java Programming", "Alice"));

bookRepository.save(new Book("Hibernate in Action","Bob"));

bookRepository.save(new Book("Microservices Guide", "Carol"));

bookRepository.save(new Book("Data Structures", "Davidraj"));

}

return "Sample books added!";

}

// Pagination + Sorting end point

@GetMapping

public Page<Book> getBooks(

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

@RequestParam(defaultValue = "3") int size,

@RequestParam(defaultValue = "title") String sortBy,

@RequestParam(defaultValue = "asc") String direction

) {

Sort sort = direction.equalsIgnoreCase("asc") ?

Sort.*by*(sortBy).ascending() :

Sort.*by*(sortBy).descending();

Pageable pageable = PageRequest.*of*(page, size, sort);

return bookRepository.findAll(pageable);

}

}

**6. BookApplication(Main Java)**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class BookApplication {

public static void main(String[] args) {

SpringApplication.*run*(BookApplication.class, args);

}

}

**OUTPUT:**

<http://localhost:8843/books/init>

Sample books added!

<http://localhost:8843/books?page=0&size=3&sortBy=title&direction=asc>

{"content":[{"id":5,"title":"Data Structures","author":"David"},{"id":3,"title":"Hibernate in Action","author":"Bob"},{"id":2,"title":"JavaProgramming","author":"Alice"}],

"pageable":{"pageNumber":0,"pageSize":3,"sort":{"sorted":true,"unsorted":false,"empty":false},"offset":0,

"paged":true,"unpaged":false},

"totalPages":2,

"totalElements":5,

"last":false,"first":true,

"numberOfElements":3,"size":3,"number":0,"sort":{"sorted":true,"unsorted":false,"empty":false},"empty":false}

<http://localhost:8843/books?page=1&size=2&sortBy=author&direction=desc>

{"content":[{"id":4,"title":"Microservices Guide","author":"Carol"},

{"id":3,"title":"Hibernate in Action","author":"Bob"}],

"pageable":{"pageNumber":1,"pageSize":2,"sort":{"sorted":true,"unsorted":false,"empty":false},

"offset":2,"paged":true,"unpaged":false},

"totalPages":3,

"totalElements":5,"last":false,"first":false,

"numberOfElements":2,

"size":2,

"number":1,"sort":{"sorted":true,"unsorted":false,"empty":false},"empty":false}