**2.SPRINGDATAJPA**

**Project structure:**

spring-data-demo/

├── pom.xml

├── src/

│ └── main/

│ ├── java/

│ │ └── com/

│ │ └── example/

│ │ └── demo/

│ │ ├── DemoApplication.java

│ │ ├── controller/

│ │ │ └── BookController.java

│ │ ├── entity/

│ │ │ ├── Author.java

│ │ │ ├── Book.java

│ │ │ └── Publisher.java

│ │ └── repository/

│ │ ├── AuthorRepository.java

│ │ ├── BookRepository.java

│ │ └── PublisherRepository.java

│ └── resources/

│ ├── application.properties

│ └── static/ (optional)

│ └── templates/ (optional)

└── target/ (generated after build)

**BookController.java**

package com.example.demo.controller;

import com.example.demo.entity.Book;

import com.example.demo.repository.BookRepository;

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

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

import java.time.LocalDate;

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

    @Autowired

    private BookRepository bookRepo;

    @GetMapping("/search")

    public List<Book> search(@RequestParam String keyword) {

        return bookRepo.findByTitleContaining(keyword);

    }

    @GetMapping("/top3")

    public List<Book> topBooks() {

        return bookRepo.findTop3ByOrderByPriceDesc();

    }

    @GetMapping("/start")

    public List<Book> startingWith(@RequestParam String prefix) {

        return bookRepo.findByTitleStartingWith(prefix);

    }

    @GetMapping("/by-date")

    public List<Book> booksByDate(@RequestParam String start, @RequestParam String end) {

        return bookRepo.findByPublishDateBetween(LocalDate.parse(start), LocalDate.parse(end));

    }

    @GetMapping("/price/greater")

    public List<Book> expensive(@RequestParam Double price) {

        return bookRepo.findByPriceGreaterThan(price);

    }

    @GetMapping("/price/less")

    public List<Book> cheaper(@RequestParam Double price) {

        return bookRepo.findByPriceLessThan(price);

    }

}

**Author.java**

package com.example.demo.entity;

import javax.persistence.\*;

import java.util.List;

@Entity

public class Author {

    @Id @GeneratedValue

    private Long id;

    private String name;

    @OneToMany(mappedBy = "author", fetch = FetchType.LAZY)

    private List<Book> books;

    public Long getId() { return id; }

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

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public List<Book> getBooks() { return books; }

    public void setBooks(List<Book> books) { this.books = books; }

}

**Book.java**

package com.example.demo.entity;

import javax.persistence.\*;

import java.time.LocalDate;

import java.util.List;

@Entity

public class Book {

    @Id @GeneratedValue

    private Long id;

    private String title;

    private LocalDate publishDate;

    private Double price;

    @ManyToOne(fetch = FetchType.EAGER)

    @JoinColumn(name = "author\_id")

    private Author author;

    @ManyToMany(fetch = FetchType.LAZY)

    @JoinTable(name = "book\_publisher",

        joinColumns = @JoinColumn(name = "book\_id"),

        inverseJoinColumns = @JoinColumn(name = "publisher\_id"))

    private List<Publisher> publishers;

    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 LocalDate getPublishDate() { return publishDate; }

    public void setPublishDate(LocalDate publishDate) { this.publishDate = publishDate; }

    public Double getPrice() { return price; }

    public void setPrice(Double price) { this.price = price; }

    public Author getAuthor() { return author; }

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

    public List<Publisher> getPublishers() { return publishers; }

    public void setPublishers(List<Publisher> publishers) { this.publishers = publishers; }

}

**Publisher.java**

package com.example.demo.entity;

import javax.persistence.\*;

import java.util.List;

@Entity

public class Publisher {

    @Id @GeneratedValue

    private Long id;

    private String name;

    @ManyToMany(mappedBy = "publishers", fetch = FetchType.LAZY)

    private List<Book> books;

    public Long getId() { return id; }

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

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public List<Book> getBooks() { return books; }

    public void setBooks(List<Book> books) { this.books = books; }

}

**AuthorRepository.java**

package com.example.demo.repository;

import com.example.demo.entity.Author;

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

public interface AuthorRepository extends JpaRepository<Author, Long> {

}

**BookRepository.java**

package com.example.demo.repository;

import com.example.demo.entity.Book;

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

import java.time.LocalDate;

import java.util.List;

public interface BookRepository extends JpaRepository<Book, Long> {

    List<Book> findByTitleContaining(String keyword);

    List<Book> findAllByOrderByPriceAsc();

    List<Book> findByTitleStartingWith(String prefix);

    List<Book> findByPublishDateBetween(LocalDate start, LocalDate end);

    List<Book> findByPriceGreaterThan(Double price);

    List<Book> findByPriceLessThan(Double price);

    List<Book> findTop3ByOrderByPriceDesc();

}

**PublisherRepository.java**

package com.example.demo.repository;

import com.example.demo.entity.Publisher;

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

public interface PublisherRepository extends JpaRepository<Publisher, Long> {

}

**DemoApplication.java**

package com.example.demo;

import com.example.demo.entity.Author;

import com.example.demo.entity.Book;

import com.example.demo.repository.AuthorRepository;

import com.example.demo.repository.BookRepository;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import java.time.LocalDate;

@SpringBootApplication

public class DemoApplication {

    public static void main(String[] args) {

        SpringApplication.run(DemoApplication.class, args);

    }

    @Bean

    CommandLineRunner runner(BookRepository bookRepo, AuthorRepository authorRepo) {

        return args -> {

            Author a1 = new Author();

            a1.setName("Leo Tolstoy");

            authorRepo.save(a1);

            Book b1 = new Book();

            b1.setTitle("War and Peace");

            b1.setPrice(600.0);

            b1.setPublishDate(LocalDate.of(2023, 1, 10));

            b1.setAuthor(a1);

            bookRepo.save(b1);

            Book b2 = new Book();

            b2.setTitle("Anna Karenina");

            b2.setPrice(450.0);

            b2.setPublishDate(LocalDate.of(2024, 3, 15));

            b2.setAuthor(a1);

            bookRepo.save(b2);

        };

    }

}

**Application.properties**

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

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.hibernate.ddl-auto=update

spring.h2.console.enabled=true

server.port=8081

**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

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

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>

    <artifactId>spring-data-demo</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>spring-data-demo</name>

    <description>Demo project for Spring Data JPA</description>

    <parent>

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

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

        <version>2.7.1</version>

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

    </parent>

    <properties>

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

    </properties>

    <dependencies>

        <!-- Spring Boot Web -->

        <dependency>

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

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

        </dependency>

        <!-- Spring Data JPA -->

        <dependency>

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

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

        </dependency>

        <!-- H2 Database -->

         <dependency>

            <groupId>com.h2database</groupId>

            <artifactId>h2</artifactId>

            <version>2.2.224</version> <!-- Updated to latest secure version -->

            <scope>runtime</scope>

        </dependency>

        <!-- Test -->

        <dependency>

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

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

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <!-- Spring Boot Maven Plugin -->

            <plugin>

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

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

            </plugin>

        </plugins>

    </build>

</project>

**Output:**

