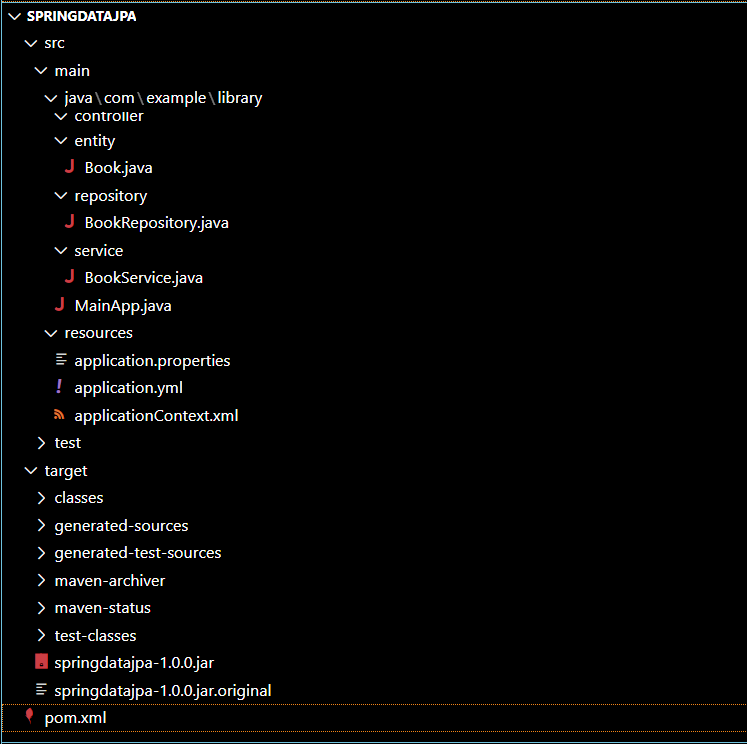
**Module 6 - Spring Data JPA with Spring Boot, Hibernate**

**Spring Data JPA - Quick Example**

**Aim:**

To build a Spring Data JPA - Quick Example

**Structure:**



package com.example.library.controller;

import com.example.library.entity.Book;

import com.example.library.service.BookService;

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

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

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

    @Autowired

    private BookService bookService;

    @GetMapping

    public List<Book> getAllBooks() {

        return bookService.getAllBooks();

    }

    @GetMapping("/{id}")

    public Book getBookById(@PathVariable Long id) {

        return bookService.getBookById(id);

    }

    @PostMapping

    public Book addBook(@RequestBody Book book) {

        return bookService.addBook(book);

    }

    @DeleteMapping("/{id}")

    public void deleteBook(@PathVariable Long id) {

        bookService.deleteBook(id);

    }

}

package com.example.library.entity;

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;

    // Constructors

    public Book() {}

    public Book(String title, String author) {

        this.title = title;

        this.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;

    }

}

package com.example.library.repository;

import com.example.library.entity.Book;

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

import org.springframework.stereotype.Repository;

@Repository

public interface BookRepository extends JpaRepository<Book, Long> {

}

package com.example.library.service;

import com.example.library.entity.Book;

import com.example.library.repository.BookRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class BookService {

    @Autowired

    private BookRepository bookRepository;

    public List<Book> getAllBooks() {

        return bookRepository.findAll();

    }

    public Book getBookById(Long id) {

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

    }

    public Book addBook(Book book) {

        return bookRepository.save(book);

    }

    public void deleteBook(Long id) {

        bookRepository.deleteById(id);

    }

}

package com.example.library;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class MainApp {

    public static void main(String[] args) {

        SpringApplication.run(MainApp.class, args);

    }

}

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

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

server.port=8081

<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>springdatajpa</artifactId>

    <version>1.0.0</version>

    <packaging>jar</packaging>

    <name>Spring Data JPA Example</name>

    <description>Spring Boot with Spring Data JPA and H2 Database</description>

    <parent>

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

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

        <version>3.2.0</version> <!-- Or your desired version -->

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

    </parent>

    <properties>

        <java.version>17</java.version> <!-- Or your installed JDK version -->

    </properties>

    <dependencies>

        <!-- Spring Boot Starter Web -->

        <dependency>

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

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

        </dependency>

        <!-- Spring Boot Starter 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>

            <scope>runtime</scope>

        </dependency>

        <!-- Spring Boot Test (Optional) -->

        <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>

                <version>3.2.0</version> <!-- Use the same as above -->

            </plugin>

        </plugins>

    </build>

</project>

**Output:**



