**SPRING BOOT**

**PRACTICE STATEMENT – 2**

**Book Class**

**package** org.statement.com;

**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** **int** id;

**private** String book\_title ;

**private** String book\_publisher;

**private** String book\_isbn;

**private** **int** book\_number\_of\_pages;

**private** **int** book\_year;

**public** Book() {}

**public** Book(**int** id, String book\_title, String book\_publisher, String book\_isbn, **int** book\_number\_of\_pages,

**int** book\_year) {

**super**();

**this**.id = id;

**this**.book\_title = book\_title;

**this**.book\_publisher = book\_publisher;

**this**.book\_isbn = book\_isbn;

**this**.book\_number\_of\_pages = book\_number\_of\_pages;

**this**.book\_year = book\_year;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getBook\_title() {

**return** book\_title;

}

**public** **void** setBook\_title(String book\_title) {

**this**.book\_title = book\_title;

}

**public** String getBook\_publisher() {

**return** book\_publisher;

}

**public** **void** setBook\_publisher(String book\_publisher) {

**this**.book\_publisher = book\_publisher;

}

**public** String getBook\_isbn() {

**return** book\_isbn;

}

**public** **void** setBook\_isbn(String book\_isbn) {

**this**.book\_isbn = book\_isbn;

}

**public** **int** getBook\_number\_of\_pages() {

**return** book\_number\_of\_pages;

}

**public** **void** setBook\_number\_of\_pages(**int** book\_number\_of\_pages) {

**this**.book\_number\_of\_pages = book\_number\_of\_pages;

}

**public** **int** getBook\_year() {

**return** book\_year;

}

**public** **void** setBook\_year(**int** book\_year) {

**this**.book\_year = book\_year;

}

}

**BookStoreRestController Class**

**package** org.statement.com;

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

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

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

**import** java.util.List;

**import** java.util.Optional;

@RestController

**public** **class** BookStoreRestController {

    @Autowired

**private** BookStoreService bookService;

    @PostMapping("/books")

**public** ResponseEntity<Book> addBook(@RequestBody Book book) {

        Book addedBook = bookService.addBook(book);

**return** **new** ResponseEntity<>(addedBook, HttpStatus.***CREATED***);

    }

    @PutMapping("/books")

**public** ResponseEntity<Book> updateBook(@RequestBody Book book) {

        Book updatedBook = bookService.updateBook(book);

**if** (updatedBook != **null**) {

**return** **new** ResponseEntity<>(updatedBook, HttpStatus.***OK***);

        } **else** {

**return** **new** ResponseEntity<>(HttpStatus.***NOT\_FOUND***);

        }

    }

    @GetMapping("/books")

**public** ResponseEntity<List<Book>> getAllBooks() {

        List<Book> books = bookService.getAllBooks();

**return** **new** ResponseEntity<>(books, HttpStatus.***OK***);

    }

    @GetMapping("/books/{id}")

**public** ResponseEntity<Book> getBookById(@PathVariable **int** id) {

        Optional<Book> book = bookService.getBookById(id);

**return** book.map(value -> **new** ResponseEntity<>(value, HttpStatus.***OK***))

                   .orElseGet(() -> **new** ResponseEntity<>(HttpStatus.***NOT\_FOUND***));

    }

    @DeleteMapping("/books/{id}")

**public** ResponseEntity<Void> deleteBookById(@PathVariable **int** id) {

        Optional<Book> book = bookService.getBookById(id);

**if** (book.isPresent()) {

        bookService.deleteBookById(id);

**return** **new** ResponseEntity<>(HttpStatus.***NO\_CONTENT***);

        } **else** {

**return** **new** ResponseEntity<>(HttpStatus.***NOT\_FOUND***);

        }

    }

    @GetMapping("/books/title/{book\_title}")

**public** ResponseEntity<List<Book>> getBooksByTitle(@PathVariable String book\_title) {

        List<Book> books = bookService.getBooksByTitle(book\_title);

**return** **new** ResponseEntity<>(books, HttpStatus.***OK***);

    }

    @GetMapping("/books/publisher/{book\_publisher}")

**public** ResponseEntity<List<Book>> getBooksByPublisherLike(@PathVariable String book\_publisher) {

        List<Book> books = bookService.getBooksByPublisherLike(book\_publisher);

**return** **new** ResponseEntity<>(books, HttpStatus.***OK***);

    }

    @GetMapping("/books?year={book\_year}")

**public** ResponseEntity<List<Book>> getBooksByYear(@RequestParam **int** book\_year) {

        List<Book> books = bookService.getBooksByYear(book\_year);

**return** **new** ResponseEntity<>(books, HttpStatus.***OK***);

    }

}

**BookStoreService Class**

**package** org.statement.com;

**import** java.util.List;

**import** java.util.Optional;

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

**import** org.springframework.stereotype.Service;

@Service

**public** **class** BookStoreService {

@Autowired

**private** BookStoreRepository bookStoreRepo;

**public** Book addBook(Book book) {

**return** bookStoreRepo.save(book);

}

**public** Book updateBook(Book book) {

**if**(bookStoreRepo.existsById(book.getId())) {

**return** bookStoreRepo.save(book);

}

**return** **null**;

}

**public** List<Book> getAllBooks() {

**return** bookStoreRepo.findAll();

}

**public** Optional<Book> getBookById(**int** id) {

**return** bookStoreRepo.findById(id);

}

**public** **void** deleteBookById(Integer id) {

bookStoreRepo.deleteById(id);

}

**public** List<Book> getBooksByTitle(String book\_title) {

**return** bookStoreRepo.findByBookTitle(book\_title);

}

**public** List<Book> getBooksByPublisherLike(String book\_publisher) {

**return** bookStoreRepo.findByBookPublisherLike(book\_publisher);

}

**public** List<Book> getBooksByYear(**int** book\_year) {

**return** bookStoreRepo.findByBookYear(book\_year);

}

}

**BookStoreRepository Class**

**package** org.statement.com;

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

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

**import** org.springframework.stereotype.Repository;

**import** java.util.List;

**import** java.util.Optional;

@Repository

**public** **interface** BookStoreRepository **extends** JpaRepository<Book, Integer> {

@Override

List<Book> findAll();

@Override

Optional<Book> findById(Integer id);

@Override

**void** deleteById(Integer id);

@Query("select b from Book b where b.book\_title=?1")

List<Book> findByBookTitle(String book\_title);

@Query("select b from Book b where b.book\_publisher=?1")

List<Book> findByBookPublisherLike(String book\_publisher);

@Query("select b from Book b where b.book\_year=?1")

List<Book> findByBookYear(**int** book\_year);

}

**Statement2Application Class**

**package** org.statement.com;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** class Statement2Application {

**public** static void main(String[] args) {

SpringApplication.run(Statement2Application.class, args);

}

}

**application.properties**

spring.application.name=Statement2

spring.datasource.url= jdbc:mysql://localhost:3306/BookApplication?createDatabaseIfNotExist=true

spring.datasource.username=root

spring.datasource.password=J2EE@java

spring.jpa.hibernate.ddl-auto=update

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

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect

**pom.xml**

<?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>3.3.2</version>

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

</parent>

<groupId>org.software</groupId>

<artifactId>Statement2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Statement2</name>

<description>Statement2 project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<configuration>

<excludes>

<exclude>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

</exclude>

</excludes>

</configuration>

</plugin>

</plugins>

</build>

</project>