**Spring Boot – Practice Exercise**Submitted By: Aniket Singh (Emp ID: 2605511)

**SQL**

create database springboot;

use springboot;

create table Book\_Details(id int(10) primary key,book\_title varchar(255),book\_publisher varchar(255),book\_isbn varchar(255),book\_number\_of\_pages int(10),book\_year int(10));

insert into Book\_Details values(101,"SQL","ABC Publishers","978-1-61729-427-3",250,2018);

insert into Book\_Details values(102,"Java","XYZ Publishers","778-2-51729-927-4",390,2024);

insert into Book\_Details values(103,"Hibernate","ABC Publishers","175-3-91821-727-7",120,2007);

insert into Book\_Details values(104,"Spring","XYZ Publishers","208-4-69829-123-5",665,2024);

insert into Book\_Details values(105,"Spring Boot","ABC Publishers","258-5-61002-007-3",478,2000);

insert into Book\_Details values(106,"HTML","XYZ Publishers","127-6-81729-227-4",257,1998);

select \* from Book\_Details;

**Book.java**

**package** com.example.demo;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.GeneratedValue;

**import** jakarta.persistence.GenerationType;

**import** jakarta.persistence.Id;

**import** jakarta.persistence.Table;

@Entity

@Table(name = "Book\_Details")

**public** **class** Book {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** Integer id;

**private** String bookTitle;

**private** String bookPublisher;

**private** String bookIsbn;

**private** **int** bookNumberOfPages;

**private** **int** bookYear;

**public** Integer getId() {

**return** id;

}

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

**this**.id = id;

}

**public** String getBookTitle() {

**return** bookTitle;

}

**public** **void** setBookTitle(String bookTitle) {

**this**.bookTitle = bookTitle;

}

**public** String getBookPublisher() {

**return** bookPublisher;

}

**public** **void** setBookPublisher(String bookPublisher) {

**this**.bookPublisher = bookPublisher;

}

**public** String getBookIsbn() {

**return** bookIsbn;

}

**public** **void** setBookIsbn(String bookIsbn) {

**this**.bookIsbn = bookIsbn;

}

**public** **int** getBookNumberOfPages() {

**return** bookNumberOfPages;

}

**public** **void** setBookNumberOfPages(**int** bookNumberOfPages) {

**this**.bookNumberOfPages = bookNumberOfPages;

}

**public** **int** getBookYear() {

**return** bookYear;

}

**public** **void** setBookYear(**int** bookYear) {

**this**.bookYear = bookYear;

}

**public** Book(Integer id, String bookTitle, String bookPublisher, String bookIsbn, **int** bookNumberOfPages,

**int** bookYear) {

**super**();

**this**.id = id;

**this**.bookTitle = bookTitle;

**this**.bookPublisher = bookPublisher;

**this**.bookIsbn = bookIsbn;

**this**.bookNumberOfPages = bookNumberOfPages;

**this**.bookYear = bookYear;

}

**public** Book() {

**super**();

}

}

**BookStoreRepository.java**

**package** com.example.demo;

**import** com.example.demo.Book;

**import** java.util.List;

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

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

List<Book> findAll();

Book findById(**int** id);

**void** deleteById(**int** id);

List<Book> findByBookTitle(String bookTitle);

List<Book> findByBookPublisherLike(String bookPublisher);

List<Book> findByBookYear(**int** bookYear);

}

**BookStoreRestController.java**

**package** com.example.demo;

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

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

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

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

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

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

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

@Controller

@RequestMapping("/books")

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

@Autowired

**private** BookStoreService bookStoreService;

@GetMapping

**public** String getAllBooks(Model model) {

model.addAttribute("books", bookStoreService.getAllBooks());

**return** "books";

}

@GetMapping("/add")

**public** String addBookForm(Model model) {

model.addAttribute("book", **new** Book());

**return** "add-book";

}

@PostMapping("/save")

**public** String saveBook(@ModelAttribute("book") Book book) {

bookStoreService.addBook(book);

**return** "redirect:/books";

}

@GetMapping("/edit/{id}")

**public** String editBookForm(@PathVariable **int** id, Model model) {

model.addAttribute("book", bookStoreService.getBookById(id));

**return** "edit-book";

}

@PostMapping("/update")

**public** String updateBook(@ModelAttribute("book") Book book) {

bookStoreService.updateBook(book);

**return** "redirect:/books";

}

@GetMapping("/delete/{id}")

**public** String deleteBook(@PathVariable **int** id) {

bookStoreService.deleteBookById(id);

**return** "redirect:/books";

}

}

**BookStoreService.java**

**package** com.example.demo;

**import** java.util.List;

**import** com.example.demo.\*;

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

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

@Service

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

@Autowired

**private** BookStoreRepository bookStoreRepository;

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

**if** (book.getId() == **null**) {

book.setId(generateNewId());

}

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

}

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

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

}

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

**return** bookStoreRepository.findAll();

}

**public** Book getBookById(**int** id) {

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

}

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

bookStoreRepository.deleteById(id);

}

**public** List<Book> getBooksByTitle(String title) {

**return** bookStoreRepository.findByBookTitle(title);

}

**public** List<Book> getBooksByPublisher(String publisher) {

**return** bookStoreRepository.findByBookPublisherLike("%" + publisher + "%");

}

**public** List<Book> getBooksByYear(**int** year) {

**return** bookStoreRepository.findByBookYear(year);

}

**private** **int** generateNewId() {

**return** bookStoreRepository.findAll().stream()

.mapToInt(Book::getId)

.max()

.orElse(0) + 1;

}

}

**DemoApplication.java**

**package** com.example.demo;

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

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

@SpringBootApplication

**public** **class** DemoApplication {

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

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

}

}

**application.properties**

spring.application.name=demo

server.port=8090

management.endpoints.web.exposure.include=\*

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

spring.datasource.username=root

spring.datasource.password=1234

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

**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>3.3.3</version>

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

</parent>

<groupId>com.example</groupId>

<artifactId>springbootday1</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>demo</name>

<description>Demo 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>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>

<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-annotations -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-annotations</artifactId>

<version>2.17.1</version>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>