**Exercise 1: Online Bookstore - Setting Up RESTful Services**

package com.example.bookstoreapi;

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

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

import org.springframework.stereotype.Service;

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

import javax.persistence.\*;

import java.util.List;

// Main Application Class

@SpringBootApplication

public class BookstoreApiApplication {

public static void main(String[] args) {

SpringApplication.run(BookstoreApiApplication.class, args);

}

}

// Model Classes

@Entity

class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String isbn;

@ManyToOne

private Author author;

// Getters and Setters

// Constructors

}

@Entity

class Author {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String bio;

// Getters and Setters

// Constructors

}

@Entity

class Customer {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

// Getters and Setters

// Constructors

}

// Repository Interfaces

interface BookRepository extends JpaRepository<Book, Long> {}

interface AuthorRepository extends JpaRepository<Author, Long> {}

interface CustomerRepository extends JpaRepository<Customer, Long> {}

// Service Classes

@Service

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 saveBook(Book book) {

return bookRepository.save(book);

}

public void deleteBook(Long id) {

bookRepository.deleteById(id);

}

}

@Service

class AuthorService {

@Autowired

private AuthorRepository authorRepository;

public List<Author> getAllAuthors() {

return authorRepository.findAll();

}

public Author getAuthorById(Long id) {

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

}

public Author saveAuthor(Author author) {

return authorRepository.save(author);

}

public void deleteAuthor(Long id) {

authorRepository.deleteById(id);

}

}

@Service

class CustomerService {

@Autowired

private CustomerRepository customerRepository;

public List<Customer> getAllCustomers() {

return customerRepository.findAll();

}

public Customer getCustomerById(Long id) {

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

}

public Customer saveCustomer(Customer customer) {

return customerRepository.save(customer);

}

public void deleteCustomer(Long id) {

customerRepository.deleteById(id);

}

}

// Controller Classes

@RestController

@RequestMapping("/api/books")

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 createBook(@RequestBody Book book) {

return bookService.saveBook(book);

}

@PutMapping("/{id}")

public Book updateBook(@PathVariable Long id, @RequestBody Book book) {

book.setId(id);

return bookService.saveBook(book);

}

@DeleteMapping("/{id}")

public void deleteBook(@PathVariable Long id) {

bookService.deleteBook(id);

}

}

@RestController

@RequestMapping("/api/authors")

class AuthorController {

@Autowired

private AuthorService authorService;

@GetMapping

public List<Author> getAllAuthors() {

return authorService.getAllAuthors();

}

@GetMapping("/{id}")

public Author getAuthorById(@PathVariable Long id) {

return authorService.getAuthorById(id);

}

@PostMapping

public Author createAuthor(@RequestBody Author author) {

return authorService.saveAuthor(author);

}

@PutMapping("/{id}")

public Author updateAuthor(@PathVariable Long id, @RequestBody Author author) {

author.setId(id);

return authorService.saveAuthor(author);

}

@DeleteMapping("/{id}")

public void deleteAuthor(@PathVariable Long id) {

authorService.deleteAuthor(id);

}

}

@RestController

@RequestMapping("/api/customers")

class CustomerController {

@Autowired

private CustomerService customerService;

@GetMapping

public List<Customer> getAllCustomers() {

return customerService.getAllCustomers();

}

@GetMapping("/{id}")

public Customer getCustomerById(@PathVariable Long id) {

return customerService.getCustomerById(id);

}

@PostMapping

public Customer createCustomer(@RequestBody Customer customer) {

return customerService.saveCustomer(customer);

}

@PutMapping("/{id}")

public Customer updateCustomer(@PathVariable Long id, @RequestBody Customer customer) {

customer.setId(id);

return customerService.saveCustomer(customer);

}

@DeleteMapping("/{id}")

public void deleteCustomer(@PathVariable Long id) {

customerService.deleteCustomer(id);

}

}