@GetMapping("/{id}")

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

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

return book.map(ResponseEntity::ok).orElseGet(() -> ResponseEntity.notFound().build());

}

package com.example.bookstoreapi.service;

import com.example.bookstoreapi.model.Book;

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

import java.util.stream.Collectors;

@Service

public class BookService {

private final List<Book> books = new ArrayList<>();

// Existing methods...

public List<Book> filterBooks(String title, String author) {

return books.stream()

.filter(book -> (title == null || book.getTitle().equalsIgnoreCase(title)) &&

(author == null || book.getAuthor().equalsIgnoreCase(author)))

.collect(Collectors.toList());

}

}

@GetMapping("/search")

public List<Book> filterBooks(@RequestParam(required = false) String title,

@RequestParam(required = false) String author) {

return bookService.filterBooks(title, author);

}

package com.example.bookstoreapi.controller;

import com.example.bookstoreapi.model.Book;

import com.example.bookstoreapi.service.BookService;

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

@RequestMapping("/books")

public class BookController {

@Autowired

private BookService bookService;

@GetMapping

public List<Book> getAllBooks() {

return bookService.getAllBooks();

}

@GetMapping("/{id}")

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

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

return book.map(ResponseEntity::ok).orElseGet(() -> ResponseEntity.notFound().build());

}

@PostMapping

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

Book savedBook = bookService.addBook(book);

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

}

@PutMapping("/{id}")

public ResponseEntity<Book> updateBook(@PathVariable Long id, @RequestBody Book bookDetails) {

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

if (optionalBook.isPresent()) {

Book book = optionalBook.get();

book.setTitle(bookDetails.getTitle());

book.setAuthor(bookDetails.getAuthor());

book.setPrice(bookDetails.getPrice());

book.setIsbn(bookDetails.getIsbn());

bookService.updateBook(book);

return ResponseEntity.ok(book);

} else {

return ResponseEntity.notFound().build();

}

}

@DeleteMapping("/{id}")

public ResponseEntity<Void> deleteBook(@PathVariable Long id) {

bookService.deleteBook(id);

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

}

@GetMapping("/search")

public List<Book> filterBooks(@RequestParam(required = false) String title,

@RequestParam(required = false) String author) {

return bookService.filterBooks(title, author);

}

}