@PostMapping

@ResponseStatus(HttpStatus.CREATED)

public Book addBook(@RequestBody Book book) {

return bookService.addBook(book);

}

@ResponseStatus(HttpStatus.NOT\_FOUND)

public class BookNotFoundException extends RuntimeException {

public BookNotFoundException(String message) {

super(message);

}

}

@GetMapping("/{id}")

public Book getBookById(@PathVariable Long id) {

return bookService.getBookById(id)

.orElseThrow(() -> new BookNotFoundException("Book not found with id " + id));

}

@PostMapping

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

Book savedBook = bookService.addBook(book);

return ResponseEntity.status(HttpStatus.CREATED)

.header("Custom-Header", "BookAdded")

.body(savedBook);

}

@PutMapping("/{id}")

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

Book updatedBook = bookService.getBookById(id)

.map(book -> {

book.setTitle(bookDetails.getTitle());

book.setAuthor(bookDetails.getAuthor());

book.setPrice(bookDetails.getPrice());

book.setIsbn(bookDetails.getIsbn());

return bookService.updateBook(book);

})

.orElseThrow(() -> new BookNotFoundException("Book not found with id " + id));

return ResponseEntity.ok()

.header("Custom-Header", "BookUpdated")

.body(updatedBook);

}

@DeleteMapping("/{id}")

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

bookService.deleteBook(id);

return ResponseEntity.noContent()

.header("Custom-Header", "BookDeleted")

.build();

}

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.\*;

@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)

.orElseThrow(() -> new BookNotFoundException("Book not found with id " + id));

}

@PostMapping

@ResponseStatus(HttpStatus.CREATED)

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

Book savedBook = bookService.addBook(book);

return ResponseEntity.status(HttpStatus.CREATED)

.header("Custom-Header", "BookAdded")

.body(savedBook);

}

@PutMapping("/{id}")

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

Book updatedBook = bookService.getBookById(id)

.map(book -> {

book.setTitle(bookDetails.getTitle());

book.setAuthor(bookDetails.getAuthor());

book.setPrice(bookDetails.getPrice());

book.setIsbn(bookDetails.getIsbn());

return bookService.updateBook(book);

})

.orElseThrow(() -> new BookNotFoundException("Book not found with id " + id));

return ResponseEntity.ok()

.header("Custom-Header", "BookUpdated")

.body(updatedBook);

}

@DeleteMapping("/{id}")

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

bookService.deleteBook(id);

return ResponseEntity.noContent()

.header("Custom-Header", "BookDeleted")

.build();

}

}