BookController.java

package com.example.bookstoreapi.controllers;

import com.example.bookstoreapi.models.Book;

import com.example.bookstoreapi.services.BookService;

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

import org.springframework.http.HttpHeaders;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

private final BookService bookService;

@Autowired

public BookController(BookService bookService) {

this.bookService = bookService;

}

@GetMapping

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

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

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

}

@GetMapping("/{id}")

@ResponseStatus(HttpStatus.FOUND)

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

Book book = bookService.getBookById(id);

return new ResponseEntity<>(book, getCustomHeaders(), HttpStatus.FOUND);

}

@PostMapping

@ResponseStatus(HttpStatus.CREATED)

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

Book newBook = bookService.createBook(book);

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

}

@PutMapping("/{id}")

@ResponseStatus(HttpStatus.ACCEPTED)

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

Book updatedBook = bookService.updateBook(id, book);

return new ResponseEntity<>(updatedBook, getCustomHeaders(), HttpStatus.ACCEPTED);

}

@DeleteMapping("/{id}")

@ResponseStatus(HttpStatus.NO\_CONTENT)

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

bookService.deleteBook(id);

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

}

private HttpHeaders getCustomHeaders() {

HttpHeaders headers = new HttpHeaders();

headers.add("X-Bookstore-API", "1.0");

headers.add("X-Response-Time", String.valueOf(System.currentTimeMillis()));

return headers;

}

}

Here's what's new:

Response Status: The @ResponseStatus annotation is used to customize the HTTP status codes for each endpoint.

Custom Headers: The getCustomHeaders() method is used to add custom headers to the response. In this example, we add two custom headers: X-Bookstore-API and X-Response-Time.