**Exercise 5: Online Bookstore - Customizing Response Status and Headers**

**BookController**

package com.bookstore.api.controllers;

import com.bookstore.api.entities.Book;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

import java.util.ArrayList;

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

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

@GetMapping

@ResponseStatus(HttpStatus.OK)

public List<Book> getAllBooks() {

return books;

}

@GetMapping("/{id}")

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

Book book = books.stream()

.filter(b -> b.getId().equals(id))

.findFirst()

.orElse(null);

return book != null ? ResponseEntity.ok(book) : ResponseEntity.notFound().build();

}

@PostMapping

@ResponseStatus(HttpStatus.CREATED)

public Book createBook(@RequestBody Book book) {

books.add(book);

return book;

}

@PutMapping("/{id}")

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

Book existingBook = books.stream()

.filter(b -> b.getId().equals(id))

.findFirst()

.orElse(null);

if (existingBook != null) {

existingBook.setTitle(book.getTitle());

existingBook.setAuthor(book.getAuthor());

existingBook.setPrice(book.getPrice());

existingBook.setIsbn(book.getIsbn());

return ResponseEntity.ok(existingBook)

.header("X-Book-Updated", "true")

.build();

}

return ResponseEntity.notFound().build();

}

@DeleteMapping("/{id}")

@ResponseStatus(HttpStatus.NO\_CONTENT)

public void deleteBook(@PathVariable Long id) {

books.removeIf(book -> book.getId().equals(id));

}

}

In the above code, I have customized the HTTP response status and headers for the book management endpoints.

- I have used @ResponseStatus to customize HTTP status codes for the endpoints.

- I have used ResponseEntity to add custom headers to the response.

Note that I have used ResponseEntity to return a custom response with headers, and @ResponseStatus to set the HTTP status code for the response.