**EXERCISE 5**

**1. Customizing Response Status**

Use @ResponseStatus to Customize HTTP Status Codes:

Use the @ResponseStatus annotation to set custom HTTP status codes for different methods in your BookController.

**1.Setting a 201 Created Status for the POST Method:**

When a new book is added, the server should return a 201 Created status code.

@PostMapping

@ResponseStatus(HttpStatus.CREATED)

public Book addBook(@RequestBody Book book) {

books.add(book);

return book;

}

**2.Setting a 404 Not Found Status for the GET Method:**

When a book is not found by its ID, you can set the status to 404 Not Found.

@GetMapping("/{id}")

@ResponseStatus(HttpStatus.NOT\_FOUND)

public Book getBookById(@PathVariable Long id) {

return books.stream()

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

.findFirst()

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

}

**Custom Exception Class:**

Create a custom exception class BookNotFoundException:

package com.example.bookstoreapi.exception;

import org.springframework.http.HttpStatus;

import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(HttpStatus.NOT\_FOUND)

public class BookNotFoundException extends RuntimeException {

public BookNotFoundException(String message) {

super(message);

}

}

**2. Adding Custom Headers Using ResponseEntity**

Add Custom Headers to the Response:

You can use ResponseEntity to add custom headers to the response in your methods.

**1.Example with POST Method:**

Add a custom header to the response when a new book is created.

@PostMapping

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

books.add(book);

return ResponseEntity.status(HttpStatus.CREATED)

.header("Custom-Header", "Book Created")

.body(book);

}

* + ResponseEntity.status(HttpStatus.CREATED): Sets the status to 201 Created.
  + .header("Custom-Header", "Book Created"): Adds a custom header Custom-Header with the value "Book Created".