**WEEK 4&5**

**EXERCISE 5**

**Online Bookstore - Customizing Response Status and Headers**

**Business Scenario**

The task is to customize the HTTP response status and headers for the book management endpoints in the online bookstore application.

**Instructions**

**1. Response Status:**

**Task:** Use @ResponseStatus to customize HTTP status codes for your endpoints.

**Implementation:**

* The @ResponseStatus annotation allows you to specify the HTTP status code that should be returned for a specific method.

**Code Example:**

@ResponseStatus(HttpStatus.CREATED)

@PostMapping("/books")

public Book addBook(@RequestBody Book book) {

return bookService.addBook(book);

}

**Explanation:**

* The addBook method is mapped to the POST /books endpoint.
* The @ResponseStatus(HttpStatus.CREATED) annotation is used to set the response status to 201 Created when a new book is successfully added.
* This approach provides a clean way to directly specify the response status without the need for a ResponseEntity.

**2. Custom Headers:**

**Task:** Add custom headers to the response using ResponseEntity.

**Implementation:**

* The ResponseEntity class allows you to control the HTTP status code, headers, and body of the response.

**Code Example:**

@GetMapping("/{id}")

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

Book book = bookService.getBookById(id);

if (book != null) {

return ResponseEntity.ok()

.header("Custom-Header", "This is a custom header")

.body(book);

} else {

return ResponseEntity.status(HttpStatus.NOT\_FOUND)

.header("Custom-Error", "Book not found")

.build();

}

}

**Explanation:**

* The getBookById method is mapped to the GET /books/{id} endpoint.
* If the book is found, a 200 OK status is returned with a custom header Custom-Header.
* If the book is not found, a 404 Not Found status is returned with a custom error header Custom-Error.
* The ResponseEntity class provides full control over the HTTP response, allowing you to add headers and customize the response based on different conditions.

**Conclusion:**

By implementing these techniques, we can customize the HTTP response status and headers for your RESTful service, enhancing its flexibility and the information provided to clients.