**Exercise 3: Online Bookstore - Handling Path Variables and Query Parameters**

**Business Scenario:**

Enhance the book management endpoints to handle dynamic URLs and query parameters.

**Instructions:**

1. **Path Variables:**
   * Implement an endpoint to fetch a book by its ID using a path variable.
2. **Query Parameters:**
   * Implement an endpoint to filter books based on query parameters like title and author.

**Answers: -**

**Exercise 3: Online Bookstore - Handling Path Variables and Query Parameters**

**1. Path Variables**

**Step-by-Step Guide:**

1. **Fetch a Book by Its ID Using a Path Variable:**
   * You've already implemented this in the previous exercise, where the getBookById method handles the retrieval of a book by its ID using a path variable.

Here's a quick recap of the relevant code:

// GET: /books/{id} - Retrieve a book by its ID

@GetMapping("/{id}")

public Book getBookById(@PathVariable int id) {

return books.stream()

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

.findFirst()

.orElse(null);

}

* + In this method, the @PathVariable annotation binds the {id} in the URL to the id parameter of the method, allowing you to retrieve the specific book.

**2. Query Parameters**

1. **Filter Books Based on Query Parameters like Title and Author:**
   * Implement an endpoint to handle filtering by using query parameters.

Here’s how you can do it:

// GET: /books/search?title={title}&author={author} - Filter books by title and/or author

@GetMapping("/search")

public List<Book> searchBooks(@RequestParam(required = false) String title,

@RequestParam(required = false) String author) {

return books.stream()

.filter(book -> (title == null || book.getTitle().equalsIgnoreCase(title)) &&

(author == null || book.getAuthor().equalsIgnoreCase(author)))

.toList();

}

* + **Explanation:**
    - **@RequestParam:** This annotation is used to extract query parameters from the URL.
    - The required = false attribute makes the query parameters optional.
    - The method filters the list of books based on the presence and values of the title and author query parameters.
  + **Example Requests:**
    - To filter books by title: GET /books/search?title=Java
    - To filter books by author: GET /books/search?author=John
    - To filter books by both title and author: GET /books/search?title=Java&author=John

By enhancing your controller with these methods, our application can now dynamically retrieve books based on both path variables and query parameters, making it more versatile and user-friendly.