**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.

**Solution**

**Bookcontroller.java**

package com.example.bookstoreapi.controller;

import com.example.bookstoreapi.model.Book;

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

import java.util.ArrayList;

import java.util.List;

@RestController

@RequestMapping("/api/books")

public class BookController {

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

// GET: Retrieve all books

@GetMapping

public List<Book> getAllBooks() {

return books;

}

// GET: Retrieve a book by ID

@GetMapping("/{id}")

public Book getBookById(@PathVariable Long id) {

return books.stream().filter(book -> book.getId().equals(id)).findFirst().orElse(null);

}

// POST: Add a new book

@PostMapping

public Book createBook(@RequestBody Book book) {

books.add(book);

return book;

}

// PUT: Update an existing book

@PutMapping("/{id}")

public Book updateBook(@PathVariable Long id, @RequestBody Book updatedBook) {

for (int i = 0; i < books.size(); i++) {

Book book = books.get(i);

if (book.getId().equals(id)) {

books.set(i, updatedBook);

return updatedBook;

}

}

return null;

}

// DELETE: Delete a book by ID

@DeleteMapping("/{id}")

public String deleteBook(@PathVariable Long id) {

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

return "Book with ID " + id + " has been deleted.";

}

// GET: 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();

}

}

**Example Requests**

1. **Fetch a Book by ID**:
   * **Request**: GET /api/books/1
   * **Response**: Returns the book with ID 1.
2. **Search for Books by Title and/or Author**:
   * **Request**: GET /api/books/search?title=Spring&author=John
   * **Response**: Returns a list of books that match the title "Spring" and author "John".
   * **Request**: GET /api/books/search?title=Spring
   * **Response**: Returns a list of books with the title "Spring", regardless of the author.
   * **Request**: GET /api/books/search?author=John
   * **Response**: Returns a list of books written by "John", regardless of the title.