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

**Business Scenario:**

This exercise focuses on enhancing the RESTful endpoints in the online bookstore application by implementing dynamic URL handling using path variables and filtering books using query parameters.

**Instructions:**

1. **Path Variables:**
   * Implement an endpoint to fetch a book by its ID using a path variable. This allows users to retrieve specific book details by including the book ID directly in the URL.
2. **Query Parameters:**
   * Implement an endpoint to filter books based on query parameters such as title and author. This enables users to search for books by specifying the title or author in the query string.

**Implementation Summary:**

1. **BookController:**
   * The BookController class handles HTTP requests related to books. It uses Spring's @RestController and @RequestMapping annotations to define RESTful endpoints.
   * **Path Variable Handling:**
     + The getBookById method is mapped to a GET request with a dynamic URL containing the book ID as a path variable. This method calls the service layer to retrieve the book details.
   * **Query Parameter Handling:**
     + The filterBooks method allows users to filter books by title and author using query parameters. If the query parameters are not provided, all books are returned.
2. **BookService:**
   * The BookService class contains the business logic for managing books.
   * The getBookById method retrieves a book by its ID, leveraging Java Streams to find the book in the collection.
   * The filterBooks method filters the list of books based on the provided title and author query parameters. The method uses Streams to apply the filters dynamically, making it flexible for handling different search criteria.
3. **Book Entity:**
   * The Book class is a JPA entity representing the books table in the database. It contains attributes like id, title, author, price, and isbn, which are mapped to the corresponding columns in the table.

**Usage:**

* **Fetching a Book by ID:**
  + Endpoint: GET /api/books/{id}

Ex: GET /api/books/1 returns the book with ID 1.

* **Filtering Books by Title and/or Author:**
  + Endpoint: GET /api/books/search

Ex: GET /api/books/search?title=Java&author=Test returns books matching the specified title and author.

**Conclusion:**

This exercise shows how to enhance a RESTful API by adding support for dynamic URLs using path variables and flexible filtering using query parameters.