### **Book Information API Integration with JSON Request/Response**

#### **Objective:**

Create a Spring Boot application that integrates with a third-party Book Information API to fetch and display book details. The application should expose a RESTful endpoint that accepts JSON requests and returns JSON responses.

#### **Requirements:**

1. **Project Setup:**
   * Use Spring Boot to create a RESTful web service.
   * Use Maven or Gradle as the build tool.
2. **External Book API:**
   * Integrate with the Open Library Books API (<https://openlibrary.org/dev/docs/api/books>) to fetch book details by ISBN.
   * No API key is required for this API.
3. **Endpoints:**
   * Create a REST endpoint: GET /book
   * The endpoint should accept an ISBN in an Json request and return book details in an Json response.
4. **Request/Response Structure:**
   * Design a POJO to map the incoming Json request (see below sample request) using jackson.
   * Design a POJO to map the outgoing Json response using jackson (see below response that your api should return) .
   * Convert the external JSON response from Open Library Books API to your JSON response format.
   * Use Builder design pattern(lombok) for conversion of sample request to open api request, do the same for response as well
   * Use lombok to reduce boilerplate code

**Error Handling:**

* + Implement proper error handling for scenarios such as invalid ISBNs, missing book information, and connectivity problems.
  + Return meaningful HTTP status codes and error messages in JSON format.

1. **Configuration:**
   * **Externalize any necessary configuration properties using Spring Boot's application.properties or application.yml.**
2. **Testing:**
   * Write unit tests for the service layer using JUnit and Mockito.
   * Write integration tests for the controller layer.

#### **Deliverables:**

1. A GitHub repository link or a zip file containing the Spring Boot project.
2. A README.md file with instructions on how to set up and run the application.
3. Sample curl or Postman commands to test the API endpoints.

### **Important Note on Usage of AI Models**

#### **Integrity of the Assignment**

We value the integrity and authenticity of your work. As part of our evaluation process, it is essential that the assignment is completed by you without the assistance of AI models or automated code generation tools. This includes, but is not limited to, tools such as ChatGPT, Copilot, or any other AI-based programming aids.

#### **Policy on AI Usage**

* **Prohibited Use:** The use of AI models to generate code, design solutions, or assist in completing any part of this assignment is strictly prohibited.
* **Verification:** We reserve the right to verify the authenticity of your work. If we determine that AI tools were used, it will result in immediate disqualification from the interview process.
* **Honesty:** We trust that you will complete this assignment with honesty and integrity. This assignment is an opportunity for you to showcase your skills and knowledge genuinely.

#### **Consequences**

If it is discovered that AI models were used to complete this assignment, your submission will be disqualified, and you will be removed from consideration for the position.

#### **Encouragement to Use Your Skills**

We encourage you to rely on your own understanding and expertise to complete this task. If you have questions or need clarification about the assignment, please reach out to us directly.

Thank you for your understanding and cooperation.

Sample incoming request:

{

"book\_isbn": "1234567"

}

Sample outgoing response

{

"book\_isbn": "1234567",

"title": "lord of ring",

"publishers": ["name"],

"authors":[{

"name": "author1"

}, {

"name": "author2"

}],

"totalPages": 50,

"published\_date": "23-05-2023"

}

Sample Error Response

{

"message": " there was error",

"details": "if any"

}