Java Developer – Code Challenge

Library Management System - Part 2

Project Extension Overview

This document outlines the tasks and requirements for extending the Library Management System project. The focus of this phase is to integrate a third-party API for enhanced functionality, enforce data constraints using Spring Boot annotations, and implement validation for borrowing transactions.

Task 1: Integration with Open Library API

Objective

Enhance the `Create Book API` to fetch the author's name from the Open Library API based on the ISBN provided during book creation.

Open Library API Details

- **Base URL**: `https://openlibrary.org`
- **Endpoint**: `/api/books`
- **Parameters**:
  - `bibkeys`: ISBN of the book (e.g., `ISBN:9780451526538`)
  - `format`: Response format, set as `json`
  - `jscmd`: Command for data, set as `data`
- **Example Request**:

  ```
  curl -X GET
  "https://openlibrary.org/api/books?bibkeys=ISBN:9780451526538&format=json&
  jscmd=data"
  ```

- **Sample Response**:

```
•   {
•     "ISBN:9780451526538": {
•       "title": "1984",
•       "authors": [
•         {
•           "name": "George Orwell",
•           "url": "https://openlibrary.org/authors/OL149084A/George_Orwell"
•         }
•       ]
•     }
    }
```

Steps for Integration

1. Fetch the author's name from the API when creating a book.
2. Populate the `Author` entity in the system using the fetched name.
3. Handle cases where the API does not return author details gracefully.
4. Test the integration using Postman with the following steps:
    - Import the API cURL command into Postman.
    - Replace `<isbn>` with the actual ISBN to test various cases.

Task 2: Enforcing Data Constraints with Spring Boot Annotations

Objective

Ensure data integrity and validation at the DTO level by applying Spring Boot validation annotations.

Constraints

1. **BookDTO**
    - `ISBN`: Must be **unique**.
    - `Category`: Must be a valid enum value.
    - All fields must be **non-null**.
2. **BorrowerDTO**
    - `Email`: Must be in a valid email format.
    - All fields must be **non-null**.

example

Add annotations like `@NotNull` to the DTO fields.

Task 3: Validation for Borrow Transactions

Objective

Add a limit on the number of borrowing transactions a borrower can have, based on a configuration value.

Requirements

1. Define a maximum transaction limit in the `application.properties` file:

    ```
    borrower.transaction.limit=5
    ```

2. Before processing a borrow transaction:
    - Check how many transactions the borrower currently has in the database.
    - Compare this number against the limit defined in the configuration.
    - Reject the transaction if the limit is exceeded.

Deliverables

1. Updated source code with:
   - Integration of the Open Library API in the `Create Book API`.
   - Spring Boot annotations for validation on DTOs.
   - Borrow transaction limit validation logic.