**Library Manager API Documentation**

**Dushan Nanayakkara**

**Overview**

The Library Manager API allows you to manage books and borrowers in a library system. It provides endpoints for book management, borrower registration, and borrowing/returning books.

**Base URL**

All API requests are made to: http://your-api-url.com/api/v1

**Authentication**

If your API requires authentication, provide details on how to authenticate each request here.

**Error Handling**

Explain how your API handles errors and what the common error responses are.

**Rate Limiting**

If your API has rate limits, explain them here.

**Endpoints**

**Books**

**GET /books**

Fetches all books. You can filter based on availability and uniqueness.

**Parameters:**

* filter\_unavailable: Filter out currently borrowed books if true. Default is false.
* filter\_non\_unique: Filter out copies of the same books if true. Default is false.

**Example:**

http://localhost:8080//api/v1/books?filter\_non\_unique=true

**Response:** A list of books.

Example Response:

    "result": [

        {

            "id": 1,

            "title": "Book Title 1",

            "author": "Author 1",

            "isbn": "978-3-16-148410-0"

        },

        {

            "id": 2,

            "title": "Book Title 2",

            "author": "Author 2",

            "isbn": "978-1-40-289460-1"

        },

        {

            "id": 3,

            "title": "Zojja",

            "author": "ABC",

            "isbn": "abc1253lererd"

        },

        {

            "id": 4,

            "title": "Zojja",

            "author": "ABC",

            "isbn": "abc1253lererd"

        },

        {

            "id": 5,

            "title": "Zojja",

            "author": "ABC",

            "isbn": "abc1253lererd"

        }

    ],

    "totalBooks": 5

}

**POST /books**

Registers a new book.

**Request Body:**

* title: The title of the book.
* author: The author of the book.
* isbn: The ISBN of the book.

Example Request Body:

{

    "title": "BOOK TITLE 1",

    "author": "Author1",

    "isbn":"9781573226127"

}

**Validations:**

* Title, author, and ISBN are validated (no to be blank or null) with spring bean validation support.

**Response:** The created book.

**Borrowers**

**POST /borrower**

Registers a new borrower.

**Request Body:**

* id: The ID of the borrower.
* name: The name of the borrower.
* email: The email of the borrower.

**Response:** The created borrower.

**Borrow Records**

**POST /books/{id}/borrow**

Borrows a book.

**Parameters:**

* id: The ID of the book to be borrowed.

**Request Body:**

* bookId: The ID of the book.
* borrowerId: The ID of the borrower.
* borrowDate: The date the book was borrowed.
* returnDate: The date the book was returned.

**Response:** The created borrow record.

**POST /books/{id}/return**

Returns a book.

**Parameters:**

* id: The ID of the book to be returned.

**Request Body:**

* bookId: The ID of the book.
* borrowerId: The ID of the borrower.
* borrowDate: The date the book was borrowed.
* returnDate: The date the book was returned.

**Response:** The updated borrow record.

**Models**

**BookDTO**

* id: The ID of the book.
* title: The title of the book.
* author: The author of the book.
* isbn: The ISBN of the book.

**BookResultDTO**

* result: list of books.
* title: The title of the book.
* totalBooks: total number of books returned.

**BorrowerDTO**

* id: The ID of the borrower.
* name: The name of the borrower.
* email: The email of the borrower.

**BorrowRecordDTO**

* bookId: The ID of the book.
* borrowerId: The ID of the borrower.
* borrowDate: The date the book was borrowed.
* returnDate: The date the book was returned.