Octane

Reading Recommendation System API Specification

1 Introduction

This document specifies the requirements for a Reading Recommendation System API, which allows users to submit their reading intervals and recommends the top-rated books in the system based on the number of unique pages read by all users.

2 System Overview

The Reading Recommendation System API provides two main operations:

1. Allow users to submit an interval of starting and ending pages that they read for a specific book. Note that the user can **submit multiple intervals** for the same book.
2. Show the top five books in the system, picked based on **how many unique pages** have been read for all the users that submitted their intervals (sorted by books from the most read pages to the least read pages).

3 API Specification

3.1 Submit a User Reading Interval

3.1.1 Request

Allows users to submit their reading intervals for a specific book.

```
{
"user_id": "111",
"book_id": "1",
"start_page": 2,
"end_page": 30
}
```

3.1.2 Response

Returns a status code indicating whether the request was successful or not.

```
{
"status_code": "success"
}
```

3.2 Calculate the Most Recommended Five Books

Performance is crucial for retrieving the top recommended books efficiently

3.2.1 Request

The API allows users to get the top five recommended books in the system.

3.2.2 Response

The API returns an array of books sorted by the number of read pages in descending order.

```
[
{
"book_id": "5",
"book_name": "test1",
"num_of_pages": "143",
"num_of_read_pages": "100"
},
{
```

```
"book_id": "1",
"book_name": "test3",
"num_of_pages": "100",
"num_of_read_pages": "90"
},
...
]
```

3.3 Role-Based Authorization

The API implements role-based authorization to restrict access to certain endpoints.

3.3.1 Authentication

Only authenticated users can submit reading intervals. The API requires users to provide a valid access token in the request header.

3.3.2 Authorization

Only admin users can create and modify books. The API checks the user's role before allowing them to access certain endpoints.

3.4 Logging and Exception Handling

The API implements logging and exception handling to improve error reporting and debugging.

3.5 Seeds

create a seeding script that populates a database with random values to simulate a large amount of data for performance testing purposes.

4 Implementation Details

4.1 Architecture

The Reading Recommendation System API follows the Clean Architecture principles, consisting of four layers: Presentation, Application, Domain, and Infrastructure.

4.2 Technologies

The API is implemented using NestJS and PostgreSQL.

4.3 GitFlow

The API codebase follows the GitFlow branching model to manage the development process.

5 Installation and Running

Make sure to include .md file that explains how to install and run the project

6 Conclusion

The Reading Recommendation System API provides a simple and efficient way for users to submit their reading intervals and get recommendations for the top-rated books in the system. The API implements role-based authorization, logging and exception handling, and unit tests to ensure secure, reliable, and maintainable code.