Book Finder API Guide 📚🔍

Description

The Book Finder API enables users to search for books, retrieve book details, manage user accounts, and submit reviews. It provides endpoints for searching books by title, author, or category, retrieving book details including ratings and reviews, managing user accounts, and submitting reviews for books. With this API, developers can build applications to help users discover and review books.

Authentication

While the current version of the API does not require authentication for accessing its endpoints, future updates may introduce authentication mechanisms to enhance security and manage user access. Authentication could involve the use of access tokens or other methods to verify user identity and authorize access to protected endpoints.

Models:

Understanding the structure of each model in the API is essential for effectively utilizing the API. Below are the key models used in the Book Finder API:

Book 📖

- `title` (String): The title of the book.
- `author` (String): The author(s) of the book.
- `category` (String): The category or genre of the book.
- `description` (String): A brief description of the book.
- `rating` (Number): The average rating of the book.
- `reviews` (Array): An array of review objects associated with the book.
- `createdAt` (Date): The date and time when the book entry was created.

User 👤

- `username` (String): The username of the user.
- `email` (String): The email address of the user.
- `password` (String): The hashed password of the user.
- `createdAt` (Date): The date and time when the user account was created.

Review ✍️

- `userId` (Object ID): The ID of the user who submitted the review.
- `bookId` (Object ID): The ID of the book being reviewed.
- `rating` (Number): The rating given by the user.
- `comment` (String): The comment or review text.
- `createdAt` (Date): The date and time when the review was submitted.

Endpoint Structure

The base URL for the Book Finder API is `http://localhost:3002/api/`.

`/books` Endpoints 📚

- `GET /books`: Retrieve all books.
- `GET /books/:id`: Retrieve details of a specific book by ID.
- `GET /books/search?title={title}&author={author}&category={category}`: Search for books by title, author, or category.
- `POST /books`: Add a new book to the database.
- `PUT /books/:id`: Update details of a specific book by ID.
- `DELETE /books/:id`: Remove a book from the database by ID.

`/users` Endpoints 👤

- `GET /users`: Retrieve all users.
- `GET /users/:id`: Retrieve details of a specific user by ID.
- `POST /users/register`: Register a new user.
- `POST /users/login`: Log in an existing user.
- `PUT /users/:id`: Update details of a specific user by ID.
- `DELETE /users/:id`: Remove a user from the database by ID.

`/reviews` Endpoints ✍️

- `GET /reviews`: Retrieve all reviews.
- `GET /reviews/:id`: Retrieve details of a specific review by ID.
- `POST /reviews`: Add a new review for a book.
- `PUT /reviews/:id`: Update details of a specific review by ID.
- `DELETE /reviews/:id`: Remove a review from the database by ID.

Status Codes

- `200 OK`: Successful request.
- `201 Created`: Resource created successfully.
- `400 Bad Request`: Invalid syntax or missing parameters.
- `404 Not Found`: Requested resource not found.

- `500 Internal Server Error`: Unexpected error on the server.
- `429 Too Many Requests`: Rate limit exceeded.

Error Handling

Detailed error messages are provided in the response body to assist with troubleshooting.

API Rate Limiting

Rate limiting is implemented to maintain stability and performance. Requests are limited to 100 per IP address within a 15-minute window.