

Bookstore Management API

Table of Contents

- Features
- Installation
- Usage
- API Endpoints
- Authentication
- Curl Commands
- Importing the Postman Collection
- Allowed Users
- Testing
- Technologies Used

Features

- Add new books to the database.
- Retrieve all books or a specific book by ISBN.
- Update book details.
- Delete books from the database.
- Basic authentication to restrict access to certain endpoints.
- Unit tests for API endpoints.
- JWT token-based authentication (Bonus feature).
- **models**: Defines the schema for the Book model.
- **routes**: Handles API endpoints for authentication and books.
- **tests**: Includes test files for endpoint testing.
- **run.py**: Main entry point of the application.
- **config.py**: Stores configuration such as database credentials.

Usage

The server starts at <http://localhost:5000> by running `python run.py`. Once the server is running, you can access the defined API endpoints.

API Endpoints

Books

- **POST /api/books**: Add a new book.
- **GET /api/books**: Retrieve all books.
- **GET /api/books/**: Retrieve a specific book by ISBN.
- **PUT /api/books/**: Update book details.
- **DELETE /api/books/**: Delete a book.

Authentication

- **POST /api/login:** Authenticate users and generate JWT tokens.

Curl Commands

Execute these curl commands in your terminal to interact with the API endpoints:

Obtain Token (Login)

```
curl -X POST -H "Content-Type: application/json" -d '{"username":  
"admin","password":"admin"}' https://mushy-bathing-suit-foal.cyclic.app/api/login
```

Replace username and password with the credentials of the allowed users.

Adding a new book

```
curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer <access_token>" -d  
"{\"title\": \"Book Title\", \"author\":  
\"Author Name\", \"ISBN\": \"1234567890\", \"price\": 10.99,  
\"quantity\": 5}" https://mushy-bathing-suit-foal.cyclic.app/api/books
```

Retrieving all books

```
curl -X GET -H "Authorization: Bearer <access_token>" https://mushy-bathing-suit-foal.cyclic.app/api/books
```

Retrieving a specific book by ISBN

```
curl -X GET -H "Authorization: Bearer <access_token>" https://mushy-bathing-suit-foal.cyclic.app/api/books/:id
```

Replace :id with the actual ISBN of the book you want to retrieve.

Updating book details

```
curl -X PUT -H "Content-Type: application/json" -H "Authorization: Bearer <access_token>" -d  
"{\"title\": \"Updated Title\", \"author\":  
\"Updated Author\", \"price\": 12.99, \"quantity\": 10}" https://mushy-bathing-suit-foal.cyclic.app/api/books/:id
```

Replace :id with the actual ISBN of the book you want to update.

Deleting a book

```
curl -X DELETE -H "Authorization: Bearer <access_token>" https://mushy-bathing-suit-foal.cyclic.app/api/books/:id
```

Replace :id with the actual ISBN of the book you want to delete.

Importing the Postman Collection

To import the Bookstore Management API collection into Postman, follow these steps:

1. Download the Postman collection file by clicking [here](#).
2. Open Postman.
3. Click on the “Import” button in the top left corner of the window.
4. In the dialog that appears, click on the “Upload Files” tab.
5. Select the downloaded collection file (.jsonformat) from your computer.

6. Once the file is uploaded, click on the “Import” button to import the collection into Postman.

After importing the collection, you’ll be able to see all the API endpoints along with example requests and responses. You can then use these requests to interact with the Bookstore Management API directly from Postman.

Allowed Users

The application allows the following users to authenticate:

- username: admin, password: admin

Testing

The application includes comprehensive test suites to validate the implemented functionalities. To run the tests, use the command `python -m unit test`.

Technologies Used

- Python 3.x
- Flask
- Flask JWT Extended
- Flask PyMongo
- PyJWT
- PyMongo (for MongoDB)