20BCE1001

Rahul Sandireddy

VIT Chennai

[Sandireddyrahul@gmail.com](mailto:Sandireddyrahul@gmail.com)

[Rahulrayudu/farmwiseai (github.com)](https://github.com/Rahulrayudu/farmwiseai)

**Farmai API Documentation**

This document provides an overview of the Farmai API, which is a simple web application built using Flask and MySQL. The API allows users to perform basic CRUD (Create, Read, Update, Delete) operations on a collection of books stored in a MySQL database.

**Authentication**

The Farmai API uses token-based authentication to secure certain routes. To access authenticated routes, clients must include a valid token in the request query parameters.

**Token Generation (Login)**

Endpoint: /login

Method: POST

**Request Body:**

username: The username (e.g., 'admin').

password: The user password.

Response:

If the login is successful, a JSON response containing a JWT (JSON Web Token) is returned.

json

Copy code

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiYWRtaW4iLCJleHBpcmF0aW9uIjoiMjAyNC0wMS0yNlQwMDowMDowMC4wMDBaIn0.Ax...8Q"

}

If the login fails, a 403 Forbidden response is returned.

Token Verification (Decorator)

The token\_required decorator is applied to routes that require authentication. It checks the validity of the token and handles expired or invalid tokens.

**Routes**

**1. Home**

Endpoint: /

Method: GET

Description: Displays the home page. If the user is not logged in, it redirects to the login page.

**2. Insert Data**

Endpoint: /insert\_data

Method: POST

Description: Inserts book data into the database.

Request Body:

json

Copy code

{

"title": "Book Title",

"author": "Author Name",

"isbn": "1234567890",

"price": 19.99,

"quantity": 50

}

Authentication: Requires a valid token.

Response:

json

Copy code

{

"message": "Data inserted successfully"

}

**3. Get All Books**

Endpoint: /get\_all\_books

Method: GET

Description: Retrieves all books from the database.

Response:

json

Copy code

{

"books": [

{

"id": 1,

"title": "Book Title",

"author": "Author Name",

"isbn": "1234567890",

"price": 19.99,

"quantity": 50

},

// ... additional books

]

}

**4. Get Book by ISBN**

Endpoint: /get\_book\_by\_isbn/<string:isbn>

Method: GET

Description: Retrieves a specific book based on its ISBN.

Parameters:

isbn: The ISBN of the book.

Response:

If the book is found:

json

Copy code

{

"book": {

"id": 1,

"title": "Book Title",

"author": "Author Name",

"isbn": "1234567890",

"price": 19.99,

"quantity": 50

}

}

If the book is not found, a 404 Not Found response is returned.

**5. Update Book**

Endpoint: /update\_book/<string:isbn>

Method: PUT

Description: Updates the information of a specific book based on its ISBN.

Parameters:

isbn: The ISBN of the book.

Request Body:

json

Copy code

{

"title": "Updated Title",

"author": "Updated Author",

"price": 24.99,

"quantity": 60

}

Authentication: Requires a valid token.

Response:

json

Copy code

{

"message": "Book updated successfully"

}

**6. Delete Book**

Endpoint: /delete\_book/<string:isbn>

Method: DELETE

Description: Deletes a specific book based on its ISBN.

Parameters:

isbn: The ISBN of the book.

Authentication: Requires a valid token.

Response:

json

Copy code

{

"message": "Book deleted successfully"

}

Conclusion

The Farmai API provides a simple interface to manage a collection of books stored in a MySQL database. Clients can authenticate using a token and perform operations such as inserting, retrieving, updating, and deleting book records. Ensure to replace placeholder values, such as the secret key and password verification, with actual secure implementations before deploying the application.