Introduction

Welcome to the E-Commerce API! This API allows users to manage products, handle user accounts, and process orders for an online shopping experience.

Getting Started

* **Base URL**: http://localhost:5000/api
* **Authentication**: To access certain endpoints, you will need a JWT token. Obtain it by logging in with valid user credentials.

API Endpoints

User Management

1. **Register User**
   * **Endpoint**: POST /users/register
   * **Request Body**:

json

{

"username": "testuser",

"email": "testuser@example.com",

"password": "securepassword"

}

* + **Response**:
    - Success: 201 Created with message.
    - Error: 400 Bad Request with error message.

1. **Login User**
   * **Endpoint**: POST /users/login
   * **Request Body**:

json

{

"email": "testuser@example.com",

"password": "securepassword"

}

* + **Response**:
    - Success: 200 OK with JWT token.
    - Error: 400 Bad Request with error message.

1. **View Profile**
   * **Endpoint**: GET /users/{id}
   * **Response**:
     + Success: 200 OK with user profile data.
     + Error: 404 Not Found if user not found.
2. **Update Profile**
   * **Endpoint**: PUT /users/{id}
   * **Request Body**:

json

{

"username": "updateduser",

"email": "updateduser@example.com"

}

* + **Response**:
    - Success: 200 OK with updated user data.
    - Error: 404 Not Found if user not found.

Product Management

1. **List Products**
   * **Endpoint**: GET /products
   * **Response**:
     + Success: 200 OK with an array of products.
2. **View Single Product**
   * **Endpoint**: GET /products/{id}
   * **Response**:
     + Success: 200 OK with product details.
     + Error: 404 Not Found if product not found.
3. **Add Product (Admin only)**
   * **Endpoint**: POST /products
   * **Request Body**:

json

{

"name": "Sample Product",

"description": "This is a sample product.",

"price": 19.99,

"stock": 100

}

* + **Response**:
    - Success: 201 Created with created product data.
    - Error: 403 Forbidden if not authorized.

1. **Update Product (Admin only)**
   * **Endpoint**: PUT /products/{id}
   * **Request Body**:

json

{

"name": "Updated Product",

"description": "Updated description.",

"price": 24.99,

"stock": 50

}

* + **Response**:
    - Success: 200 OK with updated product data.
    - Error: 404 Not Found if product not found.

1. **Delete Product (Admin only)**
   * **Endpoint**: DELETE /products/{id}
   * **Response**:
     + Success: 200 OK with success message.
     + Error: 404 Not Found if product not found.

Order Management

1. **Place Order**
   * **Endpoint**: POST /orders
   * **Request Body**:

json

{

"userId": "{user\_id}",

"items": [

{

"productId": "{product\_id}",

"quantity": 2

}

]

}

* + **Response**:
    - Success: 201 Created with order details.

1. **View Order History**
   * **Endpoint**: GET /orders
   * **Response**:
     + Success: 200 OK with an array of orders for the user.
2. **View Single Order**
   * **Endpoint**: GET /orders/{id}
   * **Response**:
     + Success: 200 OK with order details.
     + Error: 404 Not Found if order not found.
3. **Cancel Order**
   * **Endpoint**: DELETE /orders/{id}
   * **Response**:
     + Success: 200 OK with success message.
     + Error: 404 Not Found if order not found.

Error Handling

Common error responses include:

* 400 Bad Request
* 401 Unauthorized
* 403 Forbidden
* 404 Not Found
* 500 Internal Server Error

Conclusion

This guide provides a comprehensive overview of how to interact with the E-Commerce API. For any questions or issues, please reach out to the development team.