

E-Commerce Product Management API

Introduction

This Flask-based RESTful API integrates with SQLite to manage products within an e-commerce platform. Offering a range of endpoints, it facilitates product creation, retrieval, updating, and deletion. Additionally, the API boasts JWT-based authentication for enhanced security, pagination for efficient data handling, and robust validation mechanisms to ensure the integrity of user-provided information.

Requirements

- Python 3.0
- Flask
- Flask-SQLAlchemy
- Flask-JWT-Extended
- marshmallow

Installation

1. Clone the repository or download the source code.
2. Create a virtual environment and activate it.
3. Install the required dependencies using pip:

```
pip freeze > requirements.txt  
pip install -r requirements.txt
```

Configuration

1. Open the app.py file.
2. Replace 'your-secret-key' with a strong and secure secret key for JWT authentication:
`app.config['JWT_SECRET_KEY'] = 'your-strong-secret-key'`

Running the Application

1. Navigate To The Project Directory.
2. Run The Flask Application:
`python app.py`

The Application Will Start Running On [Http://localhost:5000/](http://localhost:5000/)

PRODUCTS DATABASE TABLE

Column Name	Data Type
1. id	uuid
2. Title	varchar
3. Description	varchar
4. Price	float

SQL ▾ < 1 / 1 > 1 - 5 of 5

id	title	description	price
1	mouse	wired mouse	1200.0
3	Memory	Random Access Memory (RAM)	200.0
4	Solid State Drive	Solid State Drive (SSD)	150.0
5	Motherboard	Mainboard or Motherboard	300.0
6	Motherboard	Mainboard or Motherboard	300.0

API Endpoints

Authentication

POST /login

URL: `http://localhost:5000/login`

Generates a JWT access token for authentication.

Request Body:

```
{ "username": "your_username", "password": "your_password" }
```

Response:

```
{  
  
  "access_token": "eyJ0eXAiOiJKV..."  
}
```

Products

1. Functionality : Get a product

Method : GET

Description : Retrieves a list of all products from the database. Supports pagination using the limit and skip query parameters.

Query Parameters:

limit (optional, default: 20): The maximum number of products to return.

skip (optional, default: 0): The number of products to skip or offset from the beginning of the result set.

Headers: Authorization: Bearer <access_token>

Sample Response :



2. Functionality:

POST a product

Method: POST

Description: Creates a new product with the provided details.

Headers: Authorization: Bearer

Sample Response:

Eg url : <http://localhost:5000/product>

Request Body:



ErrorResponse(400BadRequest):

```
{"error": "No Product Data Provided"}
```

3. Functionality :

Delete Product

Method: DELETE

Description: Delete a product by its id.

Headers: Authorization: Bearer <access_token>

SampleResponse:

Eg url : <http://localhost:5000/products>

Response:200 ok

The screenshot shows a REST client interface with the following details:

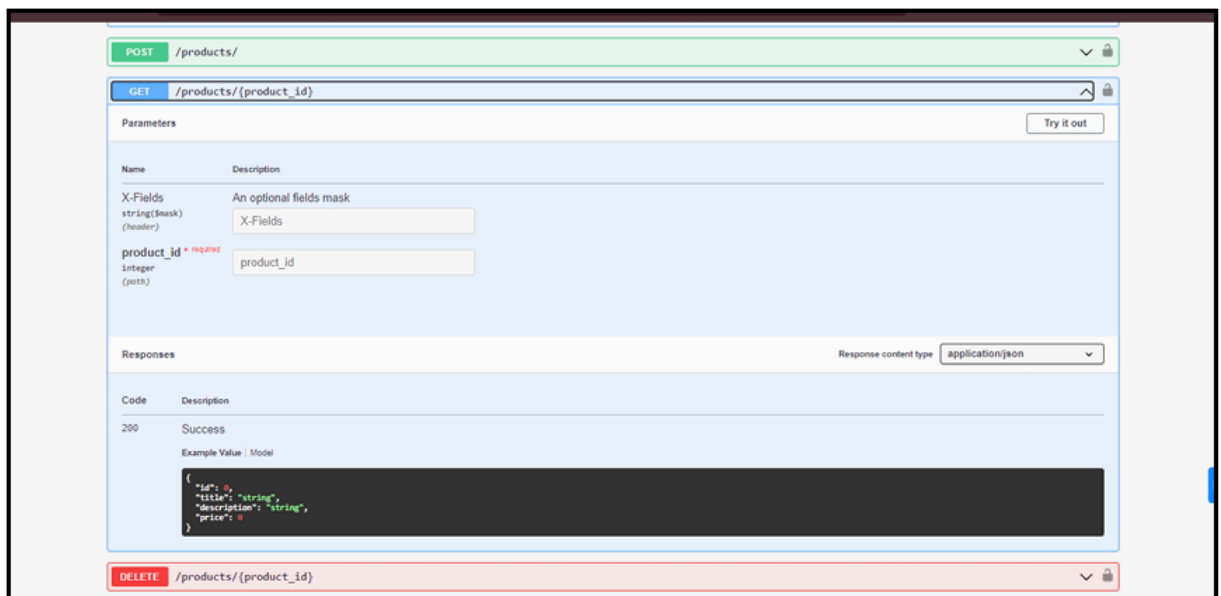
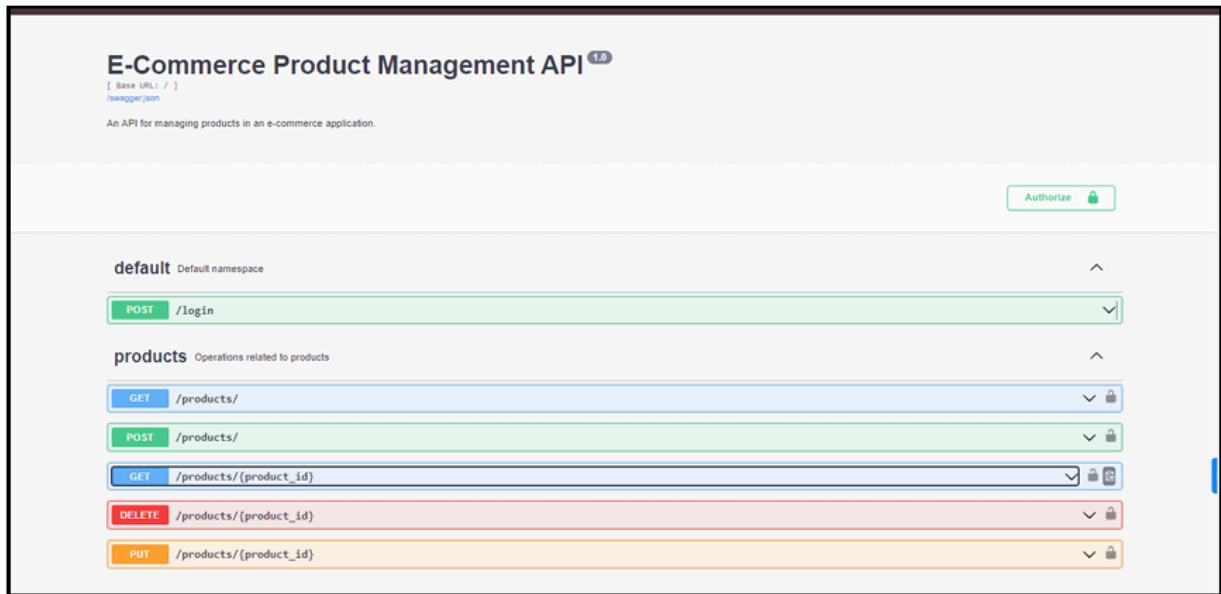
- METHOD:** DELETE
- URL:** http://localhost:5000/products/7
- HEADERS:** Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6I...
- BODY:** XHR does not allow payloads for DELETE request.
- Response:** 200 OK
- Response Headers:** Server: Werkzeug/3.0.2 Python/3.12.0, Date: Fri, 03 May 2024 22:07:52 GMT, Content-Type: application/json, Content-Length: 48 bytes, Connection: close.
- Response Body:** {"message": "Product deleted successfully"}

Error Response (404NotFound) :

The screenshot shows a REST client interface with the following details:

- METHOD:** DELETE
- URL:** http://localhost:5000/products/8
- HEADERS:** Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6I...
- BODY:** XHR does not allow payloads for DELETE request.
- Response:** 404 NOT FOUND
- Response Headers:** Server: Werkzeug/3.0.2 Python/3.12.0, Date: Fri, 03 May 2024 22:09:01 GMT, Content-Type: application/json, Content-Length: 35 bytes, Connection: close.
- Response Body:** {"error": "Product not found"}

Swagger API Documentation



Document created by : Dishank Shekokare
Mobile No. : 7741833605
Email : dishankzatrix250802@gmail.com