

E-Commerce Product Management API

Introduction

This is a RESTful API built with Flask and SQLite for managing products in an e-commerce application. The API provides endpoints for creating, retrieving, updating, and deleting products, and it also includes JWT-based authentication, pagination, and validation for user-provided data.

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 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/>.

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

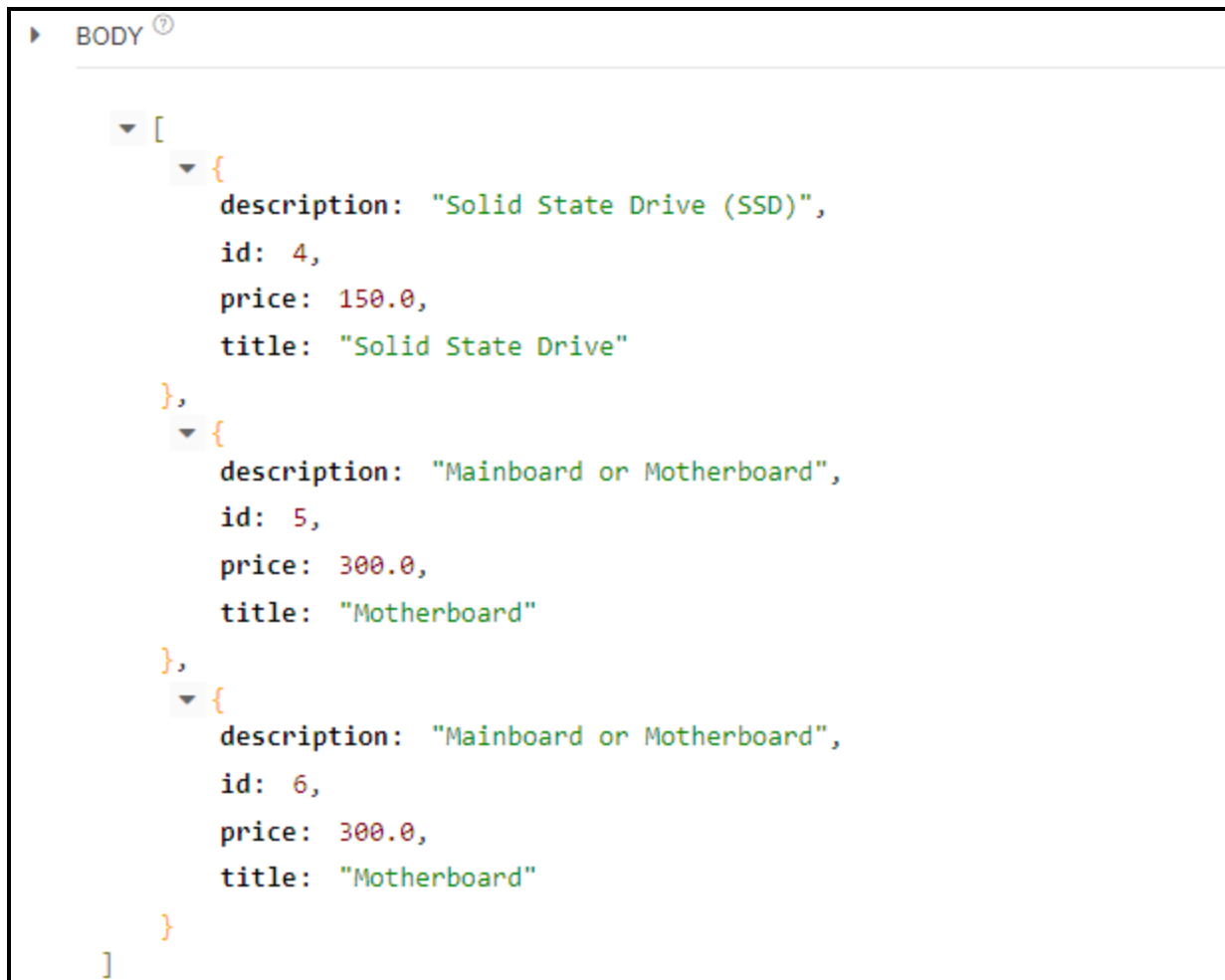
Description: Retrieves a list of all products from the database. Supports pagination using the limit and skip query parameters.

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 <access_token>

Sample Response:

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

Request Body:

```
▶ BODY ⓘ  
1 {  
2   "title": "Motherboard",  
3   "description": "Mainboard or Motherboard",  
4   "price": 300  
5 }  
6
```

Error Response (400 Bad Request):

```
{ "error": "No product data provided" }
```

3. **Functionality:** Delete a product

Method: DELETE

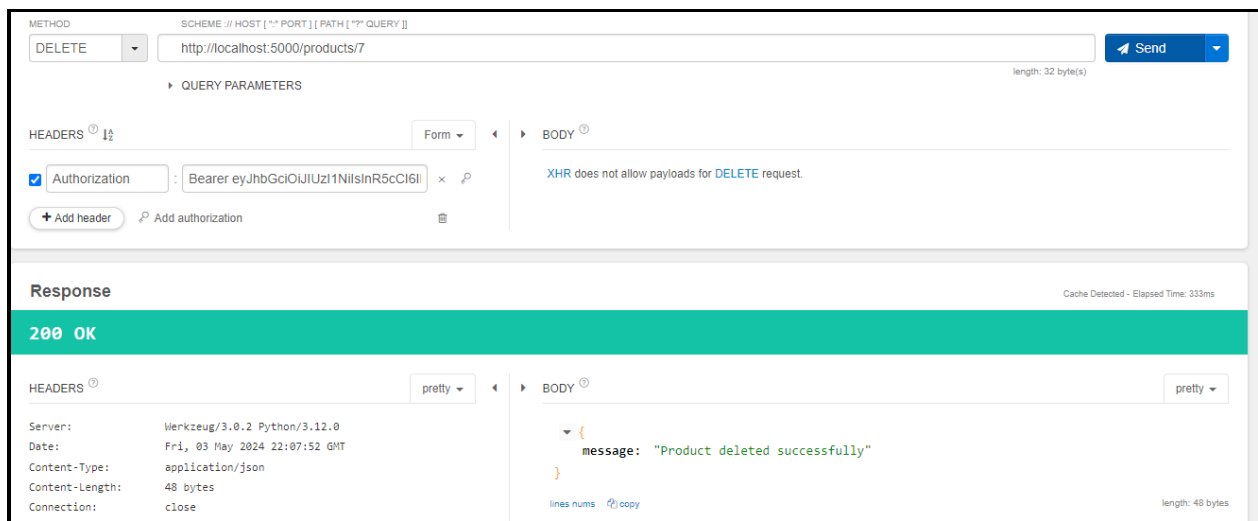
Description: Delete a product by its id.

Headers: Authorization: Bearer <access_token>

Sample Response:

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

Response:200 ok



Error Response (404 Not Found):

METHOD: DELETE | SCHEME: // | HOST: localhost:5000 | PATH: /products/8 | QUERY: | length: 32 byte(s) | Send

QUERY PARAMETERS

HEADERS: 12 | Form | BODY: XHR does not allow payloads for DELETE request.

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6Iml

+ Add header | Add authorization

Response | Cache Detected - Elapsed Time: 310ms

404 NOT FOUND

HEADERS: pretty | BODY: pretty

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

error: "Product not found"

length: 35 bytes

Swagger API Documentation

E-Commerce Product Management API ^{4.0}

[Base URL: /]
/swagger.json

An API for managing products in an e-commerce application.

Authorize

default Default namespace

POST /login

products Operations related to products

GET /products/

POST /products/

GET /products/{product_id}

DELETE /products/{product_id}

PUT /products/{product_id}

POST

/products/

GET

/products/{product_id}

Parameters

Try it out

Name	Description
X-Fields string(\$mask) (header)	An optional fields mask <input type="text" value="X-Fields"/>
product_id * required integer (path)	<input type="text" value="product_id"/>

Responses

Response content type: application/json

Code	Description
200	Success

Example Value | Model

```
{
  "id": 0,
  "title": "string",
  "description": "string",
  "price": 0
}
```

DELETE

/products/{product_id}

Document created by Himanshu Parashar
8860776298
himanshuparashar0101@gmail.com