**Configuração Inicial**

Requisitos: Python 3.x, Flask, JWT.

Instalação do Flask: Execute o comando pip install flask para instalar o Flask.

Instalação do JWT: Execute o comando pip install flask\_jwt\_extended para instalar o JWT.

Instalação do PyJWT: Execute o comando pip install PyJWT para instalar o PyJWT.

Execução: Para iniciar a API, execute o arquivo principal com python main.py.

Para visualização dos resultados é recomendado o Postman.

**Endpoints da API:**

- localhost:5000/produtos (GET)

- localhost:5000/produtos/{id} (GET)

- localhost:5000/produtos/create (POST)

- localhost:5000/produtos/edit/{id} (PUT)

- localhost:5000/produtos/delete/{id} (DELETE)

**Listando todas as tarefas**

**1. Consultar (Todos os produtos)**

Endpoint: localhost:5000/produtos (GET)

Descrição: Retorna uma lista de todos os produtos cadastrados.

Respostas:

200 OK: Sucesso.

Corpo da Resposta:

[

    {

        "data": "10/04/2024",

        "dataCompra": "15/04/2024",

        "id": 1,

        "nome": "Senhor dos Anéis - A Sociedade do Anel",

        "preco": "R$ 200,00",

        "tipo": "Livro"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "15/04/2024",

        "id": 2,

        "nome": "A História sem Fim",

        "preco": "R$ 150,00",

        "tipo": "Livro"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "17/04/2024",

        "id": 3,

        "nome": "Caneta Bic",

        "preco": "R$ 7,00",

        "tipo": "Material Escolar"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "14/04/2024",

        "id": 4,

        "nome": "Relógio de Parede",

        "preco": "R$ 25,00",

        "tipo": "Decoração"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "13/04/2024",

        "id": 5,

        "nome": "Delineador",

        "preco": "R$ 40,00",

        "tipo": "Maquiagem"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "20/04/2024",

        "id": 6,

        "nome": "Teclado",

        "preco": "R$ 100,00",

        "tipo": "Periféricos"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "20/04/2024",

        "id": 7,

        "nome": "Mouse",

        "preco": "R$ 70,00",

        "tipo": "Preiféricos"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "20/04/2024",

        "id": 8,

        "nome": "Headset",

        "preco": "R$ 150,00",

        "tipo": "Preiféricos"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "11/04/2024",

        "id": 9,

        "nome": "Brincos",

        "preco": "R$ 10,00",

        "tipo": "Bijuteria"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "11/04/2024",

        "id": 10,

        "nome": "Colar",

        "preco": "R$ 15,00",

        "tipo": "Bijuteria"

    }

]

**2. Consultar (Filtrando por id)**

Endpoint: localhost:5000/produtos/{id} (GET)

Descrição: Retorna apenas um produto cadastrado com o id descrito no Endpoint.

Respostas:

200 OK: Sucesso.

Corpo da Resposta:

(Exemplo utilizando o id 1)  
localhost:5000/produtos/1

{

    "data": "10/04/2024",

    "dataCompra": "15/04/2024",

    "id": 1,

    "nome": "Senhor dos Anéis - A Sociedade do Anel",

    "preco": "R$ 200,00",

    "tipo": "Livro"

}

(Exemplo utilizando o id 8)  
localhost:5000/produtos/8

{

    "data": "10/04/2024",

    "dataCompra": "20/04/2024",

    "id": 8,

    "nome": "Headset",

    "preco": "R$ 150,00",

    "tipo": "Preiféricos"

}

**3. Criar um novo Produto**

Endpoint: localhost:5000/produtos/create (POST)

Descrição: Retorna uma lista de todos os produtos cadastrados e o produto novo.

Respostas:

200 OK: Sucesso.

Corpo da Resposta:

Adicionando o seguinte produto:

{

    "data": "21/04/2024",

    "dataCompra": "-",

    "id": 11,

    "nome": "Camisa social",

    "preco": "R$ 90,00",

    "tipo": "Roupas"

}

Assim retornando:

[

    {Todos os produtos presentes na base descritos na primeira tarefa},

    {

        "data": "21/04/2024",

        "dataCompra": "-",

        "id": 11,

        "nome": "Camisa social",

        "preco": "R$ 90,00",

        "tipo": "Roupas"

    }

]

**4. Editar um produto**

Endpoint: localhost:5000/produtos/edit/{id} (PUT)

Descrição: Retorna a atualização do produto alterado com as novas informações.

Respostas:

200 OK: Sucesso.

Corpo da Resposta:

Exemplo alterando o produto 10:  
localhost:5000/produtos/edit/10

{

    "data": "10/04/2024",

    "dataCompra": "15/04/2024",

    "id": 10,

    "nome": "Pulseira",

    "preco": "R$ 25,00",

    "tipo": "Bijuteria"

}

Retornando o produto com as informações novas.

**5. Excluir um produto**

Endpoint: localhost:5000/produtos/delete/{id} (DELETE)

Descrição: Retorna todos os produtos menos o produto excluído utilizando o id do mesmo.

Respostas:

200 OK: Sucesso.

Corpo da Resposta:

(Excluindo o produto novo adicionado na tarefa 3)

localhost:5000/produtos/delete/11

[

    {

        "data": "10/04/2024",

        "dataCompra": "15/04/2024",

        "id": 1,

        "nome": "Senhor dos Anéis - A Sociedade do Anel",

        "preco": "R$ 200,00",

        "tipo": "Livro"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "15/04/2024",

        "id": 2,

        "nome": "A História sem Fim",

        "preco": "R$ 150,00",

        "tipo": "Livro"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "17/04/2024",

        "id": 3,

        "nome": "Caneta Bic",

        "preco": "R$ 7,00",

        "tipo": "Material Escolar"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "14/04/2024",

        "id": 4,

        "nome": "Relógio de Parede",

        "preco": "R$ 25,00",

        "tipo": "Decoração"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "13/04/2024",

        "id": 5,

        "nome": "Delineador",

        "preco": "R$ 40,00",

        "tipo": "Maquiagem"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "20/04/2024",

        "id": 6,

        "nome": "Teclado",

        "preco": "R$ 100,00",

        "tipo": "Periféricos"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "20/04/2024",

        "id": 7,

        "nome": "Mouse",

        "preco": "R$ 70,00",

        "tipo": "Preiféricos"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "20/04/2024",

        "id": 8,

        "nome": "Headset",

        "preco": "R$ 150,00",

        "tipo": "Preiféricos"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "11/04/2024",

        "id": 9,

        "nome": "Brincos",

        "preco": "R$ 10,00",

        "tipo": "Bijuteria"

    },

    {

        "data": "10/04/2024",

        "dataCompra": "15/04/2024",

        "id": 10,

        "nome": "Pulseira",

        "preco": "R$ 25,00",

        "tipo": "Bijuteria"

    }

]

**6. Caso de Method Not Allowed**

Ao utilizar um dos Endpoints descritos para outro tipo de requisição o retorno será de um 405.

Respostas:

405 Method Not Allowed

Exemplo em HTML:

<!doctype html>

<html lang=en>

<title>405 Method Not Allowed</title>

<h1>Method Not Allowed</h1>

<p>The method is not allowed for the requested URL.</p>

**7. Utilizei JWT gerando um Token simétrico para a parte de segurança.**

Utilizando o Token simétrico eu simulei um caso de erro na validação do Token acarretando na finalização da API.

Para fazer o teste do Token inválido remova o “#” da declaração da variável fake\_token na linha 129 :

# Teste de Token invalidado (Tirar o comentário para o teste)

#fake\_token = token.replace('X4','X0')