Api de dados de vitivinicultura da Embrapa

API de dados de informações vinicultura da Embrapa

**Repositório GitHub: https://github.com/Mlet40/vitivinicultura/**

Participações:

Fábio Vieira Dias - rm356356

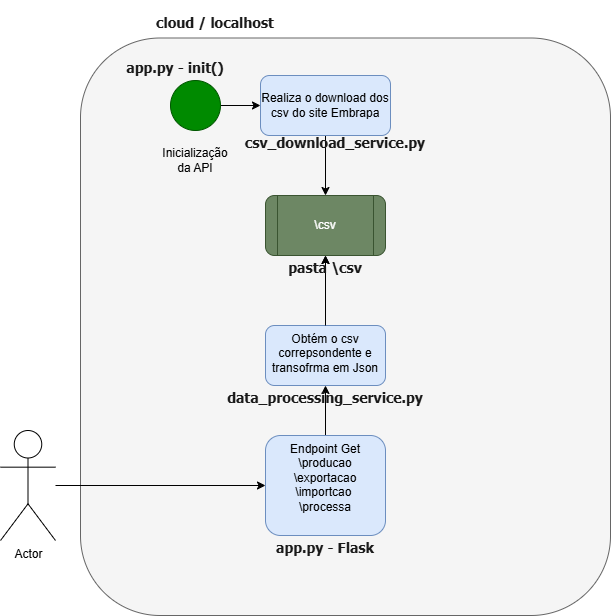
Gabriel Novaga

Cristina Caesar

# Descrição

A API de dados de informações vinicultura da Embrapa, site http://vitibrasil.cnpuv.embrapa.br/index.php?opcao=opt\_01, que é projetada para gerenciar e fornecer informações sobre a produção, processamento, comércio, importação e exportação de vinhos. Esta API é utilizada para alimentar uma base de dados que será usada para treinar modelos de machine learning, visando otimizar e prever diferentes aspectos da indústria vinícola

# Desenho Arquitetônico



**Descrição:**

Ao inicializar o app.py a aplicação irá realizar o download dos .CSVs do site Embrapa e guardará na pasta \csv. Ao consumir os recursos a aplicação irá consumir os CSVs e responderá para o cliente no formato json.

# Funcionalidades

Obter informações sobre a produção de vinhos.

Obter informações sobre o processamento de viniferas.

Obter informações sobre o comércio de vinhos.

Obter informações sobre a importação de vinhos.

Obter informações sobre a exportação de vinhos.

Autenticar usuários e fornecer tokens JWT para acesso seguro.

# Tecnologias Utilizadas

- Python

- Flask

- Flask-JWT-Extended

- OpenAPI/Swagger

- Pandas

- Gunicorn

# Configuração e Execução

Pré-requisitos

Python 3.8 ou superior

Pip (gerenciador de pacotes do Python)

# Instalação na máquina local

Siga os seguintes passos:

1- Abra o bash / cmd na pasta aonde deseja salvar o repositório

2- git clone https://github.com/Mlet40/vitivinicultura.git

3- cd vitivinicultura

4- python3 -m venv venv source venv/bin/activate # No Windows, use `venv\Scripts\activate`

5- pip install -r requirements.txt

6- Executando a API

7- python app.py

8- Para acessar e testar a api temos o swagger UI paraa facilitar, acesse [**http://localhost:5000/swagger**](http://localhost:5000/swagger)**. Como a arquitetura realiza o download na inicialização da aplicação, pode ser que demore um pouco ao iniciar.**

9 - Para acessar via Postman a collection esta na pasta raiz **Ambiente Localhost - API de dados Vitivinicultura da Embrapa.postman\_collection.json** ou o swagger **swagger.yaml**

# Execução em produção

url: https://api-vinifera-d375bceb170c.herokuapp.com/

postman: vitivinicultura\**Produção - API de dados Vitivinicultura da Embrapa.postman\_collection.json**

# Plano de deploy e execução em produção

Caso deseje realizar um deploy de uma nova versão da api em produção, siga os seguintes passos:

1 - git clone https://github.com/Mlet40/vitivinicultura.git

2 - cd vitivinicultura

3 - realizar o git pull e git checkout da branch develop

3 - Após as modificações realizar o commit e push

4 - Realizar o pull request para branch main

5 - Assim que aprovado, será automaticamente feito o deploy no heroku

6- executar a api na url: https://api-vinifera-d375bceb170c.herokuapp.com/

postman: vitivinicultura\**Produção - API de dados Vitivinicultura da Embrapa.postman\_collection.json**

swagger de produção: **https://api-vinifera-d375bceb170c.herokuapp.com/swagger**

# Testando a API

1 - Executar o endpoint /login

username:"test",

password:"test"

2- Se estiver usando o postman, automaticamente será setado o access\_token na váriavel do coleção bearerToken, e poderá utilizar os outros endpoints.

3- Caso não estiver utilizando postman, copiar o resultado do response access\_token para utilização no header autorization bearer nos demais endpoints

# Características da API

**url base: http://localhost:5000**

**Login**

Tem a função de autênticar os usuários para terem acesso aos dados da API. Ao realizar o login é retornado o access\_token, token a ser utilizado por no header de authorização Bearer em outros métodos para autorização do consumo do recurso.

Método:

- POST /login: Autentica o usuário e retorna um token JWT.

- Requisição:

{

"username": "test",

"password": "test"

}

- Resposta:

{

"access\_token": "seu\_token\_jwt"

}

**Producao**

Retorna os dados de produção vitiviníferas da Embrapa, compilados a partir do arquivo Producao.csv. A API retorna os dados em formato JSON, correlacionando as colunas do arquivo CSV 'id', 'control', 'produto' e os anos de 1970 até 2023, que foram transformadas em propriedades JSON no seguinte formato:

{

"id": 1,

"control": "control",

"produto": "produto",

"categoria":"categoria",

"ano": 1970,

"valor": 1000

}

Observe que as colunas de anos do CSV, de 1970 até 2023, foram transformadas em linhas nas colunas 'ano' e 'valor' no Json, utilizando a função 'melt' da biblioteca Pandas. Os dados da propriedade 'categoria' no Json foram extraídos da coluna 'produto' do CSV, onde determinados valores representavam subcabeçalhos que agregavam o total dos valores das informações subsequentes. Esses valores estão em caixa alta, como por exemplo 'VINHO DE MESA', e abaixo deles estão as informações detalhadas, como 'tinto', 'branco' e 'rosado'. Os subcabeçahos em caixa alta sem valores subsquentes foram extraídos como produto e categoria.

Método:

- GET /producao: Obter todas as informações de produção de vinhos.

- Requisição:

- GET /producao HTTP/1.1

- Host: localhost:5000

- Authorization: Bearer seu\_token\_jwt

**Processa**

Retorna os dados de processamento uvas, compilados a partir dos arquivos ProcessaAmericanas.csv, ProcessaMesa.csv, ProcessaSemclass.csv e ProcessaViniferas.csv. A API retorna os dados em formato JSON, correlacionando as colunas do arquivo CSV 'id', 'control', 'cultivar' e os anos de 1970 até 2023, que foram transformadas em propriedades JSON no seguinte formato:

{

"id": 1,

"control": "control",

"cultivar": "produto",

"categoria": "categoria",

"tipo": "tipo",

"ano": 1970,

"valor": 1000

}

Observe que as colunas de anos do CSV, de 1970 até 2023, foram transformadas em linhas nas colunas 'ano' e 'valor', utilizando a função 'melt' da biblioteca Pandas.Os dados da propriedade 'categoria' no Json foram extraídos da coluna 'produto' do CSV, onde determinados valores representavam subcabeçalhos que agregavam o total dos valores das informações subsequentes. Esses valores estão em caixa alta, como por exemplo 'VINHO DE MESA', e abaixo deles estão as informações detalhadas, como 'tinto', 'branco' e 'rosado'. Os subcabeçahos em caixa alta sem valores subsquentes foram extraídos como produto e categoria.

As informações da propriedade Tipo é a informação está relacionado de qual arquivo CSV ex. ProcessaAmericanas.csv, o tipo = "Americanas".

metodo:

- GET /processa: Obter todas as informações de processamento de viniferas.

- Requisição:

- GET /processa-viniferas HTTP/1.1

- Host: localhost:5000

- Authorization: Bearer seu\_token\_jwt

**Comercio**

Retorna os dados de comércio vitiviníferas da Embrapa, compilados a partir do arquivo Comercio.csv. A API retorna os dados em formato JSON, correlacionando as colunas do arquivo CSV 'id', 'control', 'produto' e os anos de 1970 até 2023, que foram transformadas em propriedades JSON no seguinte formato:

{

"id": 1,

"control": "control",

"produto": "produto",

"categoria": "categoria",

"ano": 1970,

"valor": 1000

}

Observe que as colunas de anos do CSV, de 1970 até 2023, foram transformadas em linhas nas colunas 'ano' e 'valor', utilizando a função 'melt' da biblioteca Pandas."

Os dados da propriedade 'categoria' no Json foram extraídos da coluna 'produto' do CSV, onde determinados valores representavam subcabeçalhos que agregavam o total dos valores das informações subsequentes. Esses valores estão em caixa alta, como por exemplo 'VINHO DE MESA', e abaixo deles estão as informações detalhadas, como 'tinto', 'branco' e 'rosado'. Os subcabeçahos em caixa alta sem valores subsquentes foram extraídos como produto e categoria.

metodo:

- GET /comercio: Obter todas as informações de comércio de vinhos.

- Requisição:

- GET /comercio HTTP/1.1

- Host: localhost:5000

- Authorization: Bearer seu\_token\_jwt

**Importação**

Retorna os dados de importação de derivados uvas, compilados a partir dos arquivos ImpEspumantes.csv, ImpFrescas.csv, ImpPassas.csv, ImpSuco.csv e ImpVinhos.csv. A API retorna os dados em formato JSON, correlacionando as colunas do arquivo CSV 'id', 'País', e os anos de 1970 até 2023, que foram transformadas em propriedades JSON no seguinte formato:

{

"id": 1,

"pais": "Brasil",

"ano": 1970,

"valor": 1000

}

Observe que as colunas de anos do CSV, de 1970 até 2023, foram transformadas em linhas nas colunas 'ano' e 'valor', utilizando a função 'melt' da biblioteca Pandas. Os anos duplicados no CSV foram agrupados em um só ano, somando os valores.

método:

- GET /importacao: Obter todas as informações de importação de vinhos.

- Requisição:

- GET /importacao-vinhos HTTP/1.1

- Host: localhost:5000

- Authorization: Bearer seu\_token\_jwt

**Exportacao Vinhos**

Retorna os dados de exportaçãode derivados uvas, compilados a partir dos arquivos ExpEspumantes.csv, ExpSuco.csv, ExpUva.csv e ExpVinho. A API retorna os dados em formato JSON, correlacionando as colunas do arquivo CSV 'id', 'País', e os anos de 1970 até 2023, que foram transformadas em propriedades JSON no seguinte formato:

{

"id": 1,

"pais": "Brasil",

"ano": 1970,

"valor": 1000

}

Observe que as colunas de anos do CSV, de 1970 até 2023, foram transformadas em linhas nas colunas 'ano' e 'valor', utilizando a função 'melt' da biblioteca Pandas. Os anos duplicados no CSV foram agrupados em um só ano, somando os valores.

método:

- GET /exportacao: Obter todas as informações de exportação de vinhos.

- Requisição:

- GET /exportacao-vinhos HTTP/1.1

- Host: localhost:5000

- Authorization: Bearer seu\_token\_jwt

**Alimentação da Base de Machine Learning**

Os dados fornecidos por esta API podem ser utilizados para alimentar uma base de machine learning, permitindo a criação de modelos preditivos e análises avançadas na área de vitivinicultura. A API foi desenhada para fornecer dados estruturados e facilmente integráveis em pipelines de dados para treinamento de modelos.