

Documentação do Processo de Mineração de Dados com API do GitHub

Introdução

Este documento detalha o processo de mineração de dados de repositórios GitHub utilizando a API do GitHub. Ele cobre os requisitos da API, a lógica de implementação e os critérios para referência futura, com base no código fornecido que coleta informações sobre commits, issues, pull requests e branches.

Requisitos da API

Token de Autenticação

A API do GitHub exige autenticação para acessar seus endpoints. O token de autenticação é carregado de uma variável de ambiente para segurança.

Cabeçalhos de Autenticação

Os cabeçalhos de autenticação são definidos da seguinte forma:

```
headers= {  
  'Authorization': f'token {TOKEN}',  
  'Accept': 'application/vnd.github.v3+json'  
}
```

Endpoints Utilizados

- **Commits:** `https://api.github.com/repos/{repo_name}/commits`
- **Issues:** `https://api.github.com/repos/{repo_name}/issues`
- **Pull Requests:** `https://api.github.com/repos/{repo_name}/pulls`
- **Branches:** `https://api.github.com/repos/{repo_name}/branches`

Limite de Dados da API

A API do GitHub retorna um máximo de 30 itens por página para commits, issues, pull requests e branches. Para obter todos os dados disponíveis, é necessário implementar paginação.

Funções de Mineração de Dados

Obter Nome do Repositório

Esta função extrai o nome do repositório a partir da URL fornecida.

Obter Total de Páginas

Esta função calcula o número total de páginas de dados disponíveis no endpoint específico.

Obter Dados de Todas as Páginas

Esta função obtém todos os dados de todas as páginas disponíveis para um endpoint específico.

Obter Commits

Esta função coleta informações sobre commits do repositório.

Obter Issues

Esta função coleta informações sobre issues do repositório.

Obter Pull Requests

Esta função coleta informações sobre pull requests do repositório.

Obter Branches

Esta função coleta informações sobre branches do repositório.

Interface Gráfica

A interface gráfica foi criada usando a biblioteca `customtkinter`. Ela permite que o usuário insira a URL do repositório do GitHub e selecione quais informações deseja obter (commits, issues, pull requests, branches). Os dados recuperados são exibidos na interface e salvos em um arquivo JSON.

Estrutura da Interface

A função `obter_informacoes` é responsável por coordenar a coleta de dados com base nas seleções do usuário e salvar os dados em um arquivo JSON. A interface gráfica apresenta campos para a URL do repositório e switches para selecionar os dados desejados. Após clicar no botão "Obter Informações", os dados são recuperados e exibidos na interface.

Critérios para Referência Futura

1. Autenticação
 - Sempre garantir que o token de autenticação esteja disponível e carregado corretamente das variáveis de ambiente.
2. Paginação
 - A API do GitHub retorna no máximo 30 itens por página. É essencial implementar a paginação para garantir que todos os dados sejam coletados.
3. Manipulação de Erros
 - Verificar os códigos de resposta da API e implementar manipulação de erros apropriada para garantir que o processo não falhe silenciosamente
4. Armazenamento de Dados
 - Os dados coletados devem ser armazenados em um arquivo JSON com um nome baseado na conta e no nome do repositório para fácil referência.

