

Documentação do Projeto Final - Python

Aluna: Gabrielly Simone Miranda da Silva

Curso: Python

Turma: 63730

Escola: CoderHouse

Professor: Gabriel Rodrigues

Descrição do Projeto

Este projeto final consiste em um script em Python que acessa três APIs públicas, extrai dados de interesse e os armazena em um banco de dados SQLite. A aplicação tem como objetivo integrar diferentes fontes de dados e possibilitar o armazenamento eficiente em uma base relacional.

APIs Utilizadas

O código interage com as seguintes APIs para extrair informações:

1. API de Bancos: Fornece informações sobre bancos no Brasil. Link da API: <https://brasilapi.com.br/api/banks/v1>
2. API de Feriados: Retorna informações sobre os feriados nacionais do Brasil para o ano de 2024. Link da API: <https://brasilapi.com.br/api/feriados/v1/2024>
3. API de CNPJ: Retorna dados sobre uma empresa específica, identificada por seu CNPJ. Link da API: <https://brasilapi.com.br/api/cnpj/v1/000000000000191>

Requisitos para Execução

O arquivo 'requirements.txt' contém as dependências mínimas para a execução do código. As bibliotecas necessárias são:

requests==2.31.0

pandas==1.5.3

sqlite3==3.8.3

Como Instalar as Dependências:

1. Salve o arquivo 'requirements.txt' gerado no código.
2. Execute o seguinte comando no terminal para instalar as dependências:
`pip install -r requirements.txt`

Estrutura do Código

O código está dividido nas seguintes etapas principais:

1. Importação das Bibliotecas: Importa 'requests', 'pandas' e 'sqlite3'.
2. Função 'acessar_api(url)': Função que faz uma requisição HTTP para a URL fornecida, verifica erros de requisição e retorna os dados no formato JSON.
3. Processamento dos Dados: Para cada API, os dados são extraídos e organizados em DataFrames usando a biblioteca 'pandas'.
4. Armazenamento em Banco de Dados: Os dados de cada tabela (Bancos, Feriados, CNPJ) são armazenados em um banco de dados SQLite denominado 'dados_multiplas_apis.db'.
5. Validação dos Dados Exportados: Após a exportação, o código verifica se as tabelas foram salvas corretamente no banco de dados, realizando uma consulta SQL para validar os dados.

Descrição das Funções

Função acessar_api(url)

Esta função realiza uma requisição GET para a URL fornecida e retorna os dados em formato JSON.

Parâmetros:

- url: URL da API a ser acessada.

Retorno:

- Retorna os dados em formato JSON ou None em caso de erro.

Função to_sql()

Essa função é utilizada para salvar os DataFrames processados nas tabelas do banco de dados SQLite. Cada tabela será criada ou substituída no banco de dados com os dados extraídos.

Parâmetros:

- name: Nome da tabela a ser criada.

- con: A conexão com o banco de dados.

- if_exists: Define o que fazer caso a tabela já exista. Usamos 'replace' para substituir.

- index: Define se o índice do DataFrame será gravado como uma coluna na tabela. Usamos False para não gravar o índice.

Exemplo de Saída Esperada

Após a execução do código, o banco de dados 'dados_multiplas_apis.db' será criado. As tabelas serão preenchidas com os dados obtidos das APIs. Exemplo de como o conteúdo de cada tabela pode aparecer:

Tabela de Bancos:

ISPB	Nome_Banco	Codigo_Banco	Nome_Completo
0001	BANCO DO BRASIL	001	BANCO DO BRASIL S.A.

Tabela de Feriados:

Data	Nome
01/01/2024	Confraternização Universal
07/09/2024	Independência do Brasil

Tabela de CNPJ:

cnpj	razao_social	descricao_situacao_cadastral
00000000000191	NOME DA EMPRESA LTDA	ATIVA