

Documentação do Código:

Integração de APIs e Geração de Banco de Dados

Autor: Guilherme da Cunha e Silva

Data criação: 09 de Dez. de 2024

Data última modificação: 03 de Jan. de 2025

1. Introdução

1.1 Objetivo

Este código coleta dados de três APIs distintas da [BrasilAPI](#), processa os resultados e cria um banco de dados consolidado com informações sobre bancos, corretoras e participantes do sistema PIX.

1.2 Público-alvo

O código foi desenvolvido em ambiente MacOs, no qual possui algumas especificidades do sistema operacional e destina-se a analistas de dados e/ou interessados em explorar informações financeiras e bancárias no Brasil.

1.3 Saída:

A saída é um banco de dados relacional contendo tabelas integradas e tratadas das APIs selecionadas.

2. Nível de Privacidade

Os dados manipulados são de acesso público e não contêm informações sensíveis. No entanto, o código deve ser executado em um ambiente seguro para evitar manipulações não autorizadas.

3. Pré-requisitos

3.1 Ambientes

Python 3.7 ou superior.

3.2 Bibliotecas

pandas; requests; os; time [datetime]; sqlite3.

3.3 Arquivos

Codigo_Final.ipynb - Código para execução

tabela_projeto_final_1.db - Banco de dados final do código para consulta

4. APIs Utilizadas

4.1 API de Bancos

Conexão: <https://brasilapi.com.br/api/banks/v1>

Dados fornecidos: Contém informações sobre bancos brasileiros.

4.2 API de Corretoras

Conexão: <https://brasilapi.com.br/api/cvm/corretoras/v1>

Dados fornecidos: Lista de corretoras registradas na CVM (Comissão de Valores Mobiliários).

3. API de Participantes do PIX

Conexão: <https://brasilapi.com.br/api/pix/v1/participants>

Dados fornecidos: Detalhes sobre instituições financeiras participantes do sistema PIX do dia atual ou anterior.

5. Bibliotecas Utilizadas

5.1 **pandas**: Manipulação e processamento de dados.

5.2 **requests**: Requisições HTTP para acessar as APIs.

5.3 **os**: Interações com o sistema operacional do MacOs.

5.4 **time [datetime]**: Controle de intervalos entre requisições.

5.5 **sqlite3**: Gerenciamento do banco de dados relacional.

6. Funções Criadas

6.1 Nome da Função: notificacao

Parâmetros:

url (str): URL da API que será consultada.

title (str): Título da notificação exibida no sistema.

Retorno:

- *data_json* (*dict* ou *None*): O conteúdo JSON da resposta da API caso a conexão seja bem-sucedida, ou *None* em caso de erro.

Descrição:

Esta função realiza uma requisição para uma URL especificada. Dependendo do status da conexão, exibe uma notificação no sistema informando se a conexão foi bem-sucedida ou não. Em caso de sucesso, retorna o conteúdo JSON da resposta. Caso contrário, retorna *None*.

6.2 Nome da Função: *conversor_datetime***Parâmetros:**

df (*pandas.DataFrame*): *DataFrame* contendo as colunas a serem convertidas.

colunas (*list[str]*): Lista com os nomes das colunas que devem ser convertidas para o formato *datetime*.

Retorno:

df (*pandas.DataFrame*): O *DataFrame* com as colunas especificadas convertidas para o formato *datetime*. Valores inválidos são marcados como nulos.

Descrição:

Esta função percorre uma lista de colunas de um *DataFrame* e converte os valores dessas colunas para o tipo *datetime*. Caso haja valores inválidos, estes são substituídos por nulos. A função utiliza o argumento *errors='coerce'* para evitar erros durante o processo de conversão.

6.3 Nome da Função: *conversor_int64***Parâmetros:**

df (*pandas.DataFrame*): *DataFrame* contendo as colunas a serem convertidas.

colunas (*list[str]*): Lista com os nomes das colunas que devem ser convertidas para o formato *Int64*.

Retorno:

df (pandas.DataFrame): O DataFrame com as colunas especificadas convertidas para o formato Int64. Valores inválidos são marcados como nulos.

Descrição:

Esta função converte os valores das colunas especificadas de um DataFrame para o tipo Int64. Valores que não podem ser convertidos para numéricos são transformados em nulos por meio do argumento `errors='coerce'`. Além disso, a função garante que as colunas resultantes tenham o tipo Int64 (compatível com valores nulos).

7. Tratamentos e Transformações de Dados

Validação de Resposta da conexão API: Verifica o código de status do link de cada requisição. Se a conexão for bem-sucedida (status 200), o JSON da API é processado, originando o DataFrame inicial.

Limpeza de Dados: Remoção de duplicatas, preenchimento de valores ausentes e normalização de campos para garantir consistência dos campos de cada coluna.

Estruturação de Dados: Os dados de cada API são armazenados em tabelas separadas no banco de dados relacional.

Notificações: Mensagens informam o status de cada requisição durante a execução.

8. Método de Saída

Formato: Banco de dados SQLite.

Estrutura: Tabelas individuais para *banks*, *corretoras*, *corretoras_canceladas* e *pix*. Cada tabela inclui todos os campos retornados pelos tratamentos necessários envolvendo as APIs anteriores.

9. Exemplo de Consulta

Consulta exemplo para acessar a tabela *banks* dentro do banco de dados *tabela_projeto_final_1*:

```
<# testando conexão
```

```
conn = sqlite3.connect('tabela_projeto_final_1.db')
```

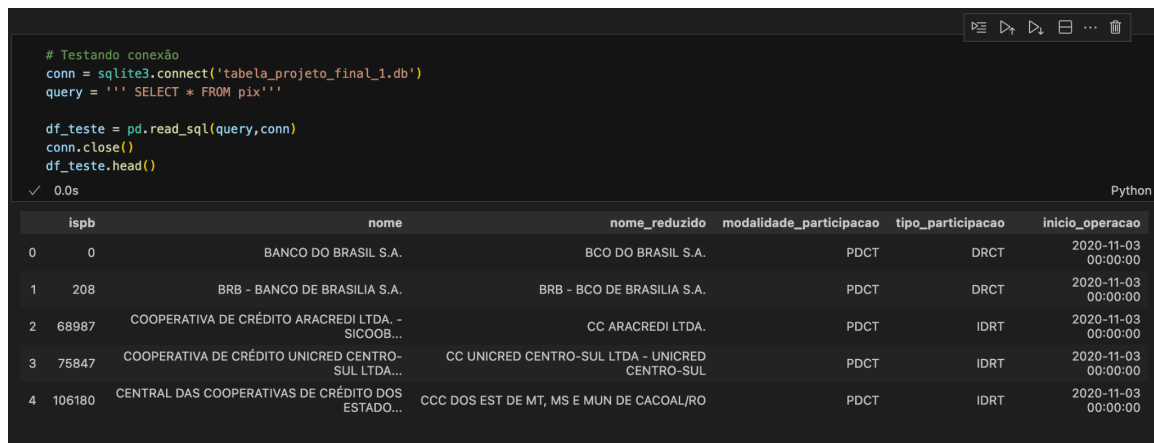
```
query = ''' SELECT * FROM pix'''
```

```
df_teste = pd.read_sql(query,conn)
```

```
conn.close()
```

```
df_teste.head()>
```

Resultado: Retorna as primeiras 5 linhas da tabela `pix`, conforme print em sequência.



The screenshot shows a Jupyter Notebook interface with a code cell containing the following Python code:

```
# Testando conexão
conn = sqlite3.connect('tabela_projeto_final_1.db')
query = ''' SELECT * FROM pix'''

df_teste = pd.read_sql(query,conn)
conn.close()
df_teste.head()
```

Below the code cell, the output shows a table with 7 columns: `ispb`, `nome`, `nome_reduzido`, `modalidade_participacao`, `tipo_participacao`, and `inicio_operacao`. The table contains 5 rows of data, representing the first 5 lines of the 'pix' table.

	ispb	nome	nome_reduzido	modalidade_participacao	tipo_participacao	inicio_operacao
0	0	BANCO DO BRASIL S.A.	BCO DO BRASIL S.A.	PDCT	DRCT	2020-11-03 00:00:00
1	208	BRB - BANCO DE BRASILIA S.A.	BRB - BCO DE BRASILIA S.A.	PDCT	DRCT	2020-11-03 00:00:00
2	68987	COOPERATIVA DE CRÉDITO ARACREDI LTDA. - SICOOB...	CC ARACREDI LTDA.	PDCT	IDRT	2020-11-03 00:00:00
3	75847	COOPERATIVA DE CRÉDITO UNICRED CENTRO-SUL LTDA...	CC UNICRED CENTRO-SUL LTDA - UNICRED CENTRO-SUL	PDCT	IDRT	2020-11-03 00:00:00
4	106180	CENTRAL DAS COOPERATIVAS DE CRÉDITO DOS ESTADO...	CCC DOS EST DE MT, MS E MUN DE CACOAL/RO	PDCT	IDRT	2020-11-03 00:00:00

10. Versionamento

Bibliotecas e versões utilizadas estão armazenadas no arquivo requirements, no formato .txt, no qual apresentam tais índices:

appnope==0.1.4

decorator==5.1.1

asttokens==3.0.0

executing==2.1.0

certifi==2024.8.30

idna==3.10

charset-normalizer==3.4.0

ipykernel==6.29.5

comm==0.2.2

ipython==8.30.0

DateTime==5.5

jedi==0.19.2

debugpy==1.8.9

jupyter_client==8.6.3

jupyter_core==5.7.2	python-dateutil==2.9.0.post0
matplotlib-inline==0.1.7	pytz==2024.2
nest-asyncio==1.6.0	pyzmq==26.2.0
numpy==2.1.3	requests==2.32.3
packaging==24.2	setuptools==75.6.0
pandas==2.2.3	six==1.16.0
parso==0.8.4	stack-data==0.6.3
pexpect==4.9.0	tornado==6.4.2
platformdirs==4.3.6	traitlets==5.14.3
prompt_toolkit==3.0.48	tzdata==2024.2
psutil==6.1.0	urllib3==2.2.3
ptyprocess==0.7.0	wcwidth==0.2.13
pure_eval==0.2.3	zope.interface==7.2
Pygments==2.18.0	

11. Referências

BrasilAPI. Documentação oficial. Disponível em: <https://brasilapi.com.br/docs>. Acesso em: 15 dez. 2024.

SQLite. Documentação oficial. Disponível em: <https://sqlite.org/docs.html>. Acesso em: 15 dez. 2024.

PANDAS. Documentação oficial. Disponível em: <https://pandas.pydata.org/docs>. Acesso em: 15 dez. 2024.

REQUESTS. Documentação oficial. Disponível em: <https://docs.python-requests.org/en/latest/>. Acesso em: 15 dez. 2024.

SQLite3 em Python. Disponível em: <https://docs.python.org/3/library/sqlite3.html>. Acesso em: 15 dez. 2024.

Real Python. Working with Datetimes in Python. Disponível em:
<https://realpython.com/python-datetime>. Acesso em: 15 dez. 2024.