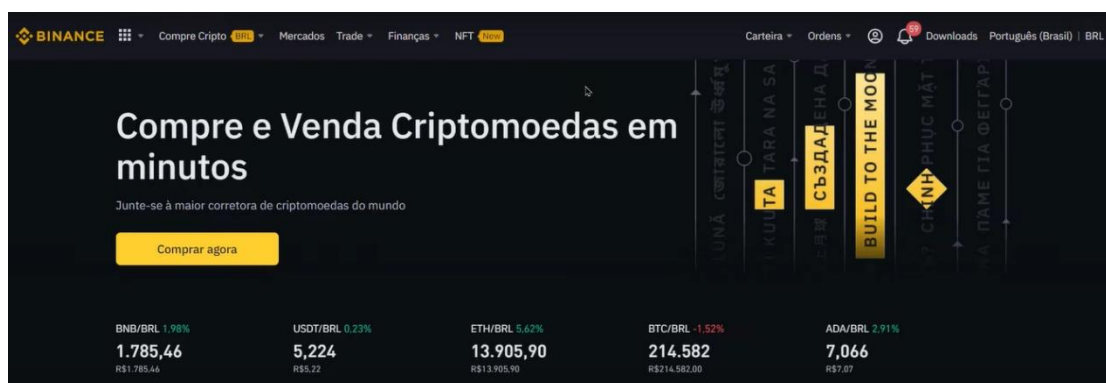


PROJETO: ETL (Extract, Transform and Load) DE UMA API (COMPLETO)

- UTILIZANDO API PARA COTAÇÃO DE MOEDAS EM PORTAIS DA WEB,
- DESENVOLVENDO SCRIPT EM PYTHON PARA CAPTURA DAS COTAÇÕES DAS MOEDAS NUM ANO,
- ENVIANDO DE TABELA DO GOOGLE COLAB PARA O GOOGLE BIGQUERY,
- ALICANDO DE QUERIES NAS TABELAS DO BIGQUERY,
- CONSTRUINDO GRÁFICOS NO GOOGLE DATA STUDIO.

1ª PARTE: CONSULTA A UM PORTAL PARA REALIZAÇÃO DE COTAÇÃO DE MOEDAS.

AS CONSULTAS PODEM SER FEITAS NOS PORTAIS API DE COTAÇÕES DE MOEDAS AWESOME OU NO BINANCE.



NESTES PORTAIS É POSSÍVEL GERENCIAR UMA CHAVE DE REGISTRO PARA A API.

PARTE 02: DESENVOLVIMENTO DE UM SCRIPT EM PYTHON PARA ACESSO A API DO PORTAL , REALIZAÇÃO DAS COTAÇÕES DE MOEDAS, E CONSTRUÇÃO DO DATAFRAME PARA ENVIO DO GOOGLE COLAB AO GCP BIGQUERY.

moedas_API.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

Arquivos

..

config

sample_data

angular-cosmos-349812-1d1fb1cf...

angular-cosmos-349812-44a96e08...

Projeto:

Cotação de moedas na API do portal awesomeapi.com.br.

Extração de dados do portal awesomeapi.com.br.

Transferência de DataFrames com cotações do google colab para o GCP BigQuery e mforma de tabelas.

Tratamento de dados das tabelas em SQL no BigQuery.

Geração de gráfico com DataStudio.

[1] import requests
import json
from datetime import datetime

from datetime import datetime
import numpy as np
import pandas as pd
import os
from google.cloud import bigquery
from google.oauth2 import service_account

cotacoes1 = requests.get("https://economia.awesomeapi.com.br/last/USD-BRL,EUR-BRL,BTC-BRL")
cotacoes1_dic = cotacoes1.json()
print(cotacoes1_dic)
cotacao_USDBRL = cotacoes1_dic['USD_BRL']['bid'] ##bid é o valor da cotação da moeda
cotacao_EURBRL = cotacoes1_dic['EUR_BRL']['bid']
cotacao_BTCBRL = cotacoes1_dic['BTC_BRL']['bid']
print("as cotações são:" "USD_BRL:" ,cotacao_USDBRL, "EURBRL:" ,cotacao_EURBRL, "BTBRL:" ,cotacao_BTCBRL)
print(type(cotacoes1_dic))

{'USD_BRL': {'code': 'USD', 'codein': 'BRL', 'name': 'Dólar Americano/Real Brasileiro', 'high': '5.1195', 'low': '5.0813', 'as cotações são:USD_BRL: 5.114 EURBRL: 5.0756 BTBRL: 110.256
<class 'dict'>}

[4] credentials=service_account.Credentials.from_service_account_file("/content/angular-cosmos-349812-44a96e086949-icnn")

[14] as cotações são:USD_BRL: 5.114 EURBRL: 5.0756 BTBRL: 110.256
<class 'dict'>

[4] credentials=service_account.Credentials.from_service_account_file("/content/angular-cosmos-349812-44a96e086949.json",
scopes=["https://www.googleapis.com/auth/cloud-platform", "https://www.googleapis.com/auth/drive"])
credentials
<google.oauth2.service_account.Credentials at 0x7fa4d45316d0>

cotacoes2 = requests.get("https://economia.awesomeapi.com.br/json/daily/USD-BRL/365?start_date=20220101&end_date=20220822")
cotacoes2_dic = cotacoes2.json()
df2 = pd.DataFrame(cotacoes2_dic, columns = ['high','low','varBid','bid','ask','timestamp'])
df2['timestamp']=pd.to_datetime(df2['timestamp'], unit='s')
print(df2.head())
print(df2.info())

high low varBid bid ask timestamp
0 5.1579 5.1571 0.0002 5.1571 5.1581 2022-08-22 21:36:41
1 5.2192 5.1648 0.0021 5.168 5.171 2022-08-19 20:59:55
2 5.2192 5.1648 0.0011 5.168 5.169 2022-08-19 20:59:30
3 5.1685 5.1681 0.0011 5.168 5.169 2022-08-18 22:24:51
4 5.1665 5.1655 0.0023 5.166 5.167 2022-08-17 23:57:00
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185 entries, 0 to 184
Data columns (total 6 columns):

0s conclusão: 10:28

The image displays three sequential screenshots of a Jupyter Notebook interface, showing the process of data processing and storage in BigQuery.

First Screenshot: The notebook shows a DataFrame with 6 columns: high, low, varBid, bid, ask, and timestamp. The data is loaded from a BigQuery table named 'cotacoes3'. The DataFrame has 185 non-null entries for each column. The dtypes are: high (object), low (object), varBid (object), bid (object), ask (object), and timestamp (datetime64[ns]).

Second Screenshot: The notebook shows the DataFrame being concatenated with another one. The code uses `df3.to_csv('dadosBigQuery.csv', index=False)` to save the data to a CSV file. The final output shows the DataFrame with 627 entries.

Third Screenshot: The notebook shows the final DataFrame with 627 entries. The dtypes are: high (object), low (object), varBid (object), bid (object), ask (object), and timestamp (datetime64[ns]). The memory usage is 29.5+ KB.

PARTE 03: APLICAÇÃO DE QUERIES EM SQL NO GCP BIGQUERY.

EXEMPLOS: FILTRAR SOMENTE A DATA NA VARIÁVEL TIMESTAMP. CONCATENAR AS TABELAS, SELECIONAR PERÍODOS DE TEMPO PARA ANÁLISE.

Google Cloud My First Project

Pesquisa Produtos, recursos, documentos (/)

Explorer

Ver projetos fixos.

- angular-cosmos-349812
 - tb02
 - tb03
 - tb04

tb04 CONSULTA COMPARTILHAR

ESQUEMA DETALHES VISUALIZAR

Filtro Insira o nome ou o valor da propriedade

Nome do campo	Tipo	Modo	Compilação	Val
high	STRING	NULLABLE		
low	STRING	NULLABLE		
varBid	STRING	NULLABLE		
bid	STRING	NULLABLE		
ask	STRING	NULLABLE		
timestamp	TIMESTAMP	NULLABLE		

EXECUTAR

```

1 SELECT bid, date(timestamp) as date FROM
2   angular-cosmos-349812.tb02
3   order by date;

```

Pressione Alt+F1 para consultar as opções de acessibilidade.

Resultados da... SALVAR RESULTADOS

INFORMAÇÕES DO JOB RESULTADOS JSON D

Linha	bid	date
1	5.6766	2022-01-03
2	5.677	2022-01-04
3	5.6708	2022-01-05

Resultados por página: 50 1 - 50 de 185

HISTÓRICO PESSOAL HISTÓRICO DO PROJETO

EDITAR ESQUEMA VER POLÍTICAS DE ACESSO DE LINHA

ATUALIZAR

PARTE 04: CONSTRUÇÃO DE UM PAINEL NO DATASTUDIO:

