Python: comandos, APIs, integração com Banco de Dados e Pandas

Poliana N Ferreira

KORU escola

Revisão

Itens importantes - Python

- Listas
- Tuplas
- Range
- Dicionários
- Funções
- Objetos multidimensionais

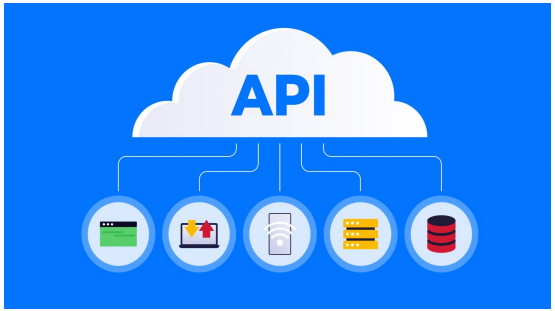
Treino de lógica de programação

https://judge.beecrowd.com/pt/

APIs

O que são APIs?

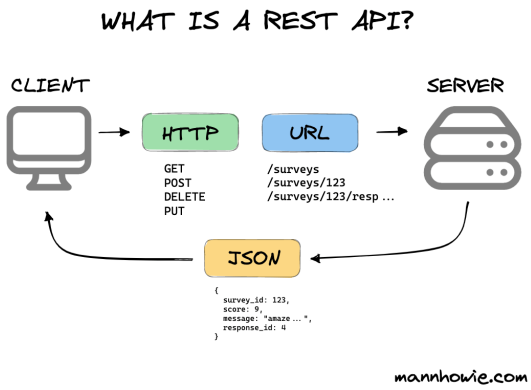
- API é um acrônimo para Interface de Programação de Aplicações;
- Uma forma simples de comunicação entre aplicações;
- Não necessita de telas, respostas são geralmente em json;
- Independente de front-end, a API não possui ligação com o front;
- Baseadas em requisição e resposta;



O que são APIs?

Conectar componentes e aplicativos em uma arquitetura de microsserviços.

O aplicativo ou serviço que está realizando o acesso é chamado de **cliente**, e o aplicativo ou serviço contendo o recurso é chamado de **servidor**.



Consumindo uma API simples

requests

API correios -

- Requisição do tipo GET
- Envia cpf via url
- Recebe o endereço

```python
import requests

# URL da API que você quer acessar
api_url = 'https://api.brasilaberto.com/v1/zipcode/01001000'

response = requests.get(api_url)
print(response.json())
```

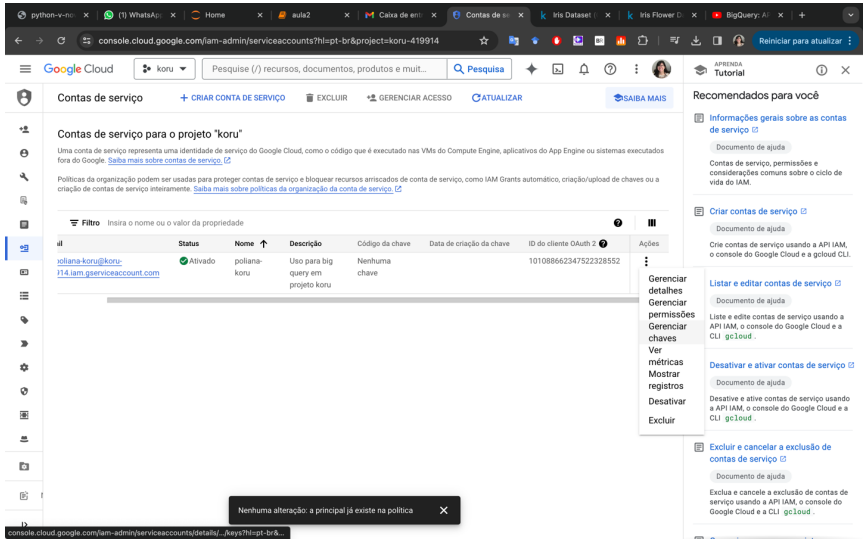
Consulta de Banco de Dados – Big Query

API do GCloud

Credenciais

Forma de “logar” no google cloud a partir do nosso computador

- IAM e administrador -> Conta de serviço
- Criar conta de serviço
- Gerenciar chaves
- Gerar json e baixar para a pasta do nosso jupyter notebook



Dados - Kaggle

Plataforma de Projetos e Dados

- Kaggle é uma plataforma de competição de ciência de dados e uma comunidade online de cientistas de dados e profissionais de aprendizado de máquina da Google LLC.

- Possui centenas de bases de dados gratuitas para testar e utilizar em projetos

- Vamos pegar a base de dados do íris e colocar no Big Query para fazermos a requisição - https://www.kaggle.com/datasets/arshid/iris-flower-dataset?resource=download

API do GCloud

Biblioteca g cloud big query

- Instalar

```
pip install google-cloud-bigquery
```

- Importar

```
from google.cloud import bigquery
from google.oauth2 import service_account
```

API do GCloud

Biblioteca g cloud big query

- Logamos usando credenciais do json

```
credentials = service_account.Credentials.from_service_account_file(filename='koru-419914-1307a43ad7ad.json',
                                                                    scopes=['https://www.googleapis.com/auth/cloud-platform'])
```

- Criamos uma conexão com o big query

```
client = bigquery.Client(credentials=credentials)
```

Colocando os dados do csv no Big Query

- Pegamos o id do projeto e definimos o nome da base de dados e da tabela que iremos criar

```
dataset_id = 'koru-419914.iristeste'
table_id = 'koru-419914.iristeste.iris'
```

- Criamos a base de dados

```
dataset = bigquery.Dataset(dataset_id)
dataset.location = "US"
dataset = client.create_dataset(dataset, exists_ok=True)
```

Colocando os dados do csv no Big Query

- Definimos o esquema da tabela

```
schema = [
    bigquery.SchemaField("species", "STRING", mode="REQUIRED"),
    bigquery.SchemaField("sepal_length", "FLOAT64", mode="REQUIRED"),
    bigquery.SchemaField("petal_length", "FLOAT64", mode="REQUIRED"),
    bigquery.SchemaField("sepal_width", "FLOAT64", mode="REQUIRED"),
    bigquery.SchemaField("petal_width", "FLOAT64", mode="REQUIRED"),
]
```

- Criamos a tabela

```
table = bigquery.Table(table_id, schema=schema)
table = client.create_table(table, exists_ok=True)
```

Colocando os dados do csv no Big Query

- Preparamos os dados a serem inseridos (pandas e o csv)

```
df = pd.read_csv('../../../Downloads/Iris.csv')
```

- Inserimos os dados

```
job = client.load_table_from_dataframe(df, table_id)
job.result()

print("Sucesso!")
```

Resgatando dados do Big Query

- Criamos a query

```
query = '''
    SELECT * FROM `iristeste.iris`
'''
```

- Fazemos a requisição

```
query_job = client.query(query)
```

```
query_job.to_dataframe()
```

Pandas

O que é Pandas?

Pandas é uma biblioteca de Python para trabalhar com dados

- Amplamente utilizada para manipulação e análise de dados
- Indispensável para cientistas de dados

Eficiente e flexível

Agora vamos avaliar a aula?

Obrigada!

escola

Consulta de Banco de Dados – sqlite3 (extra)

Biblioteca sqlite3

- Cria um arquivo .db no diretório do projeto
- Forma fácil de manter dados e usar SQL em projetos locais
- Já vem instalada no python

Biblioteca sqlite3

- Importamos a biblioteca

```
import sqlite3
```

- Conectamos com a base de dados

```
con = sqlite3.connect("db-teste2.db")
cur = con.cursor()
```

Biblioteca sqlite3

- Executamos SQL e fazemos commit

```
cur.execute("CREATE TABLE produto (nome TEXT, descricao TEXT, preco REAL, PRIMARY KEY(nome));")
cur.execute("INSERT INTO produto VALUES('macbook', 'computador apple', 15000);")
cur.execute("INSERT INTO produto VALUES('macbook2', 'computador apple', 15000);")
con.commit()
```

- Criamos lista a partir dos valores do select

```
res = cur.execute("SELECT * FROM produto;")
tabela = res.fetchall()
print(tabela)
```

- Fechamos a conexão

```
con.close()
```