kHi buddies, this post is the first part of a project that I am doing to increase my knowledge about getting and manipulating data. The idea came from a job interview that a did on my path to becoming a data scientist

No processo seletivo eu tinha um case de um problema e tinha que resolve-lo e apresentar os resultados. Passei o fim de semana resolvendo esse case (o processo começou na sexta-feira e devia ser entregue na segunda), talvez devido minha pouca experiencia foi bem difícil fazer tudo nesse tempo (foi meu primeiro processo seletivo para essa area, além disso, a empresa era de grande porte, uma emissora de televisão conhecida internacionalmente então me empenhei o máximo que pude pois queria muito passer. Spoiler: não passei hahah), pois bem, a parte em que fui mal no case foi na apresentação da resolução do problema, o “deploy”, onde o recrutador estava esperando algo mais “profissional” e com o pouco tempo que tive, apresentei um power point que até foi elogiado, porém, não era o que estava esperando. Aprendi muito fazendo esse case, e a partir dele resolve que um dia faria algo parecido. Lembro de ouvir muito dele que utilizavam a linguagem R e faziam o deploy com shiny com gráficos interativos e etc (era isso o que ele esperava do deploy). Sendo assim resolve criar um projeto como uma meta pessoal em que ia fazer todo o processo desde a extração dos dados até a apresentação, mas agora não mais em power point, mas em Power BI, Google data Studio além de um dashboard publicado na web desenvolvido com ajuda de bibliotecas do python (não é shiny ainda mas acredito que sejam semelhantes hehe).

As etapas desse projeto serão:

1º WebScrape - Extração dos dados a partir de uma API publica.

Aproveitando que estamos em ano eleitoral, resolvi pegar os dados publicos dos atuais deputados e os gastos realizados com dinheiro público desde o início da pandemia até atualmente (2020-2022).

Palavras-chave: Python, webscrape, API

2º Cloud and database integration with python.

Nesta etapa iremos criar uma conta na OracleCloudInfrastructure (OCI), onde poderemos criar um banco de dados e armazenar os dados extraídos.

Palavras-chave: Python, Oracle Cloud, Database, MySQL, Alwaysfree, VPN

3º Power BI

4º Google Data Studio

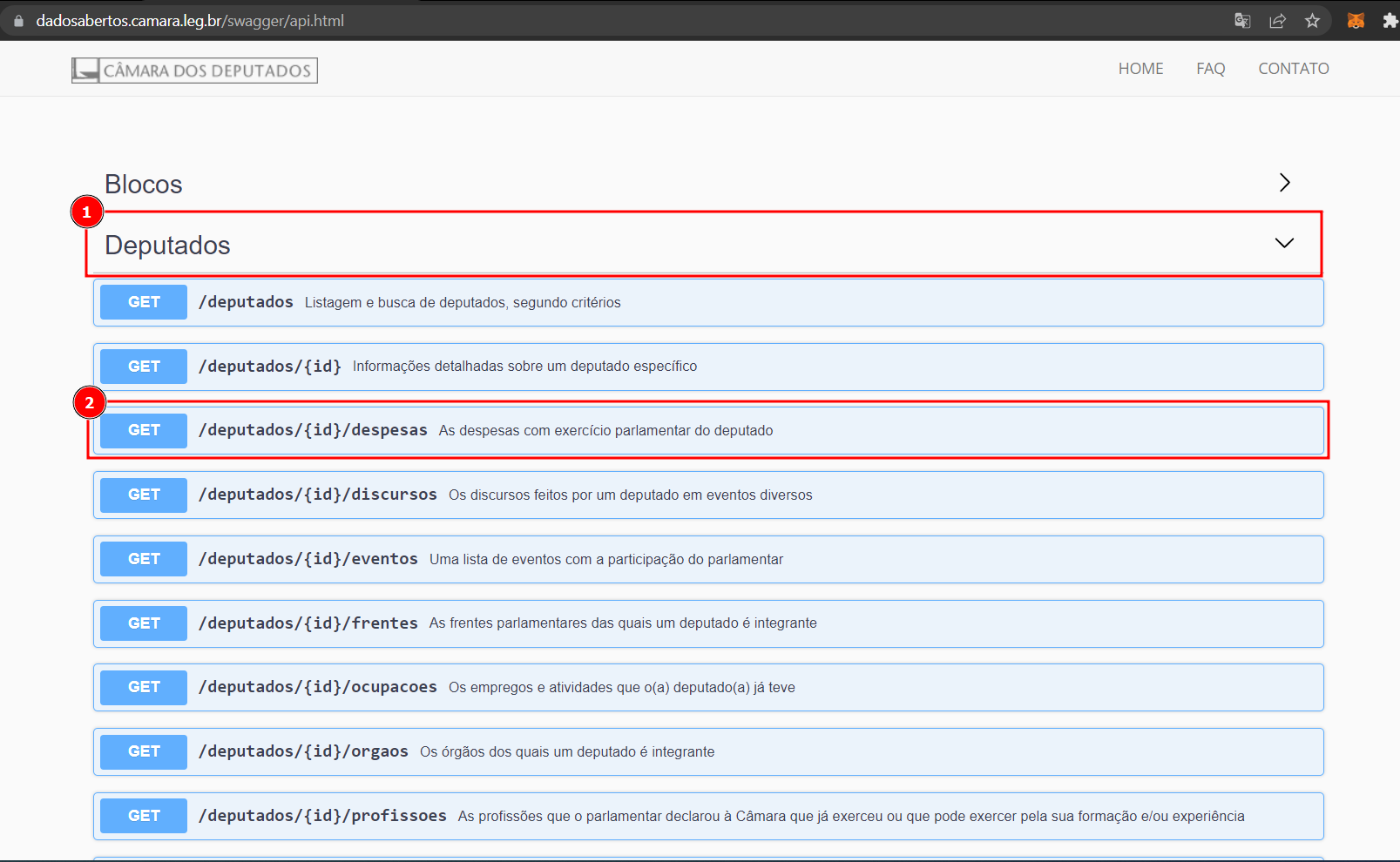
5º Python dashboard

In this section I want to show step by step how to get some data from a public API using python.

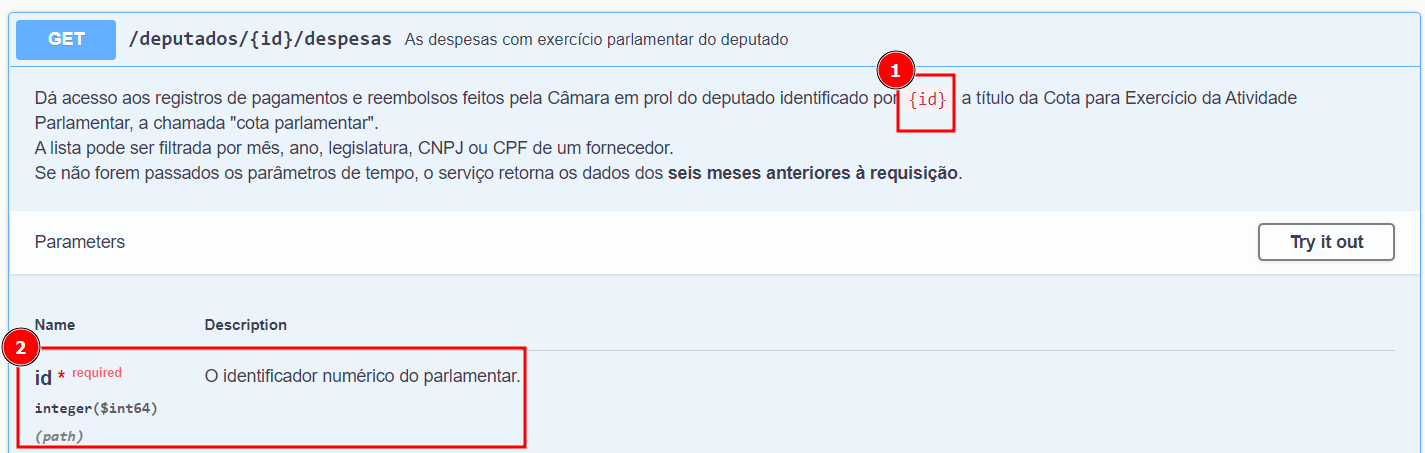
Alright, let’s get started.

Acessando a URL: <https://dadosabertos.camara.leg.br/swagger/api.html>

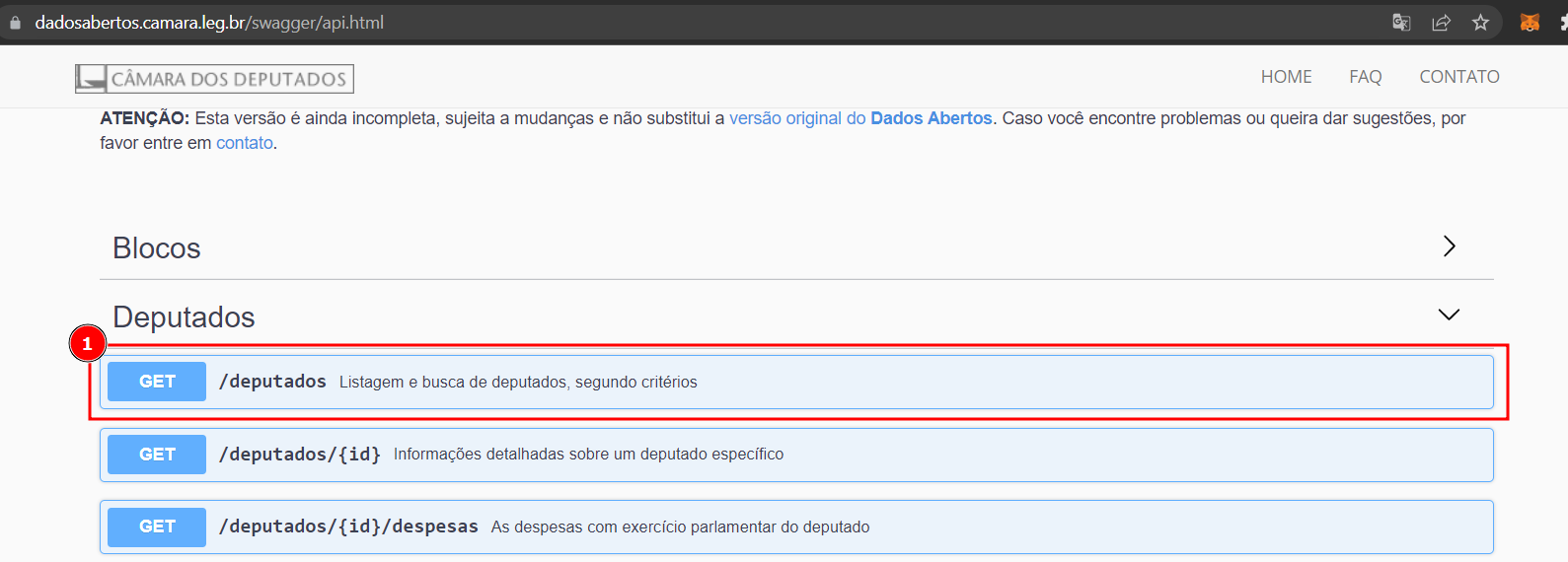
Temos a toda a documentação da API dos dados abertos. Nessa página buscamos pelas despezas do deputados:



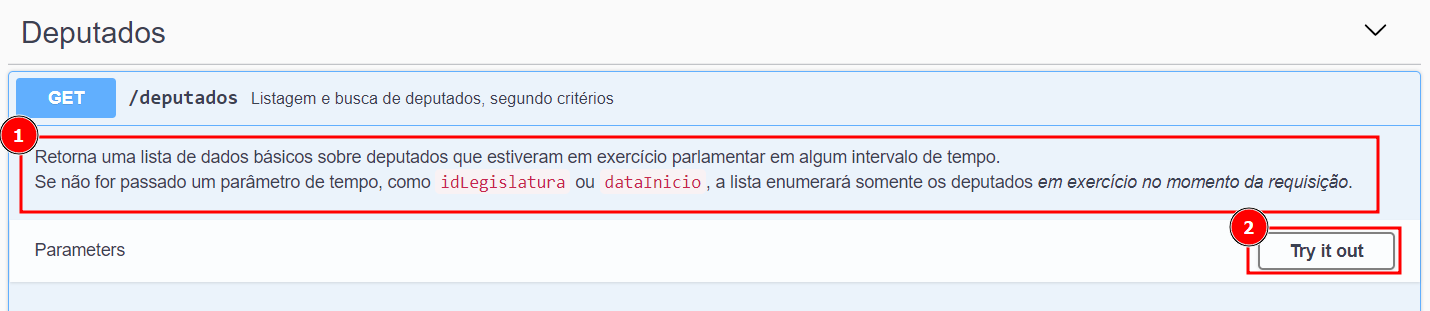
Clicando na aba de despezas, Podemos observar que para conseguirmos os dados, temos que obrigatóriamente ter o “ID” do deputado como é demonstrado na documentação.



No caso, queremos os dados das despezas de todos os deputados, então clicamos na aba de deputados pois lá Podemos encontrar a URL para assim, fazer requisições que trarão informações dos deputados e entre essas informações, os “ID’s” de cada um deles.



Clicando na aba deputados temos o seguinte:



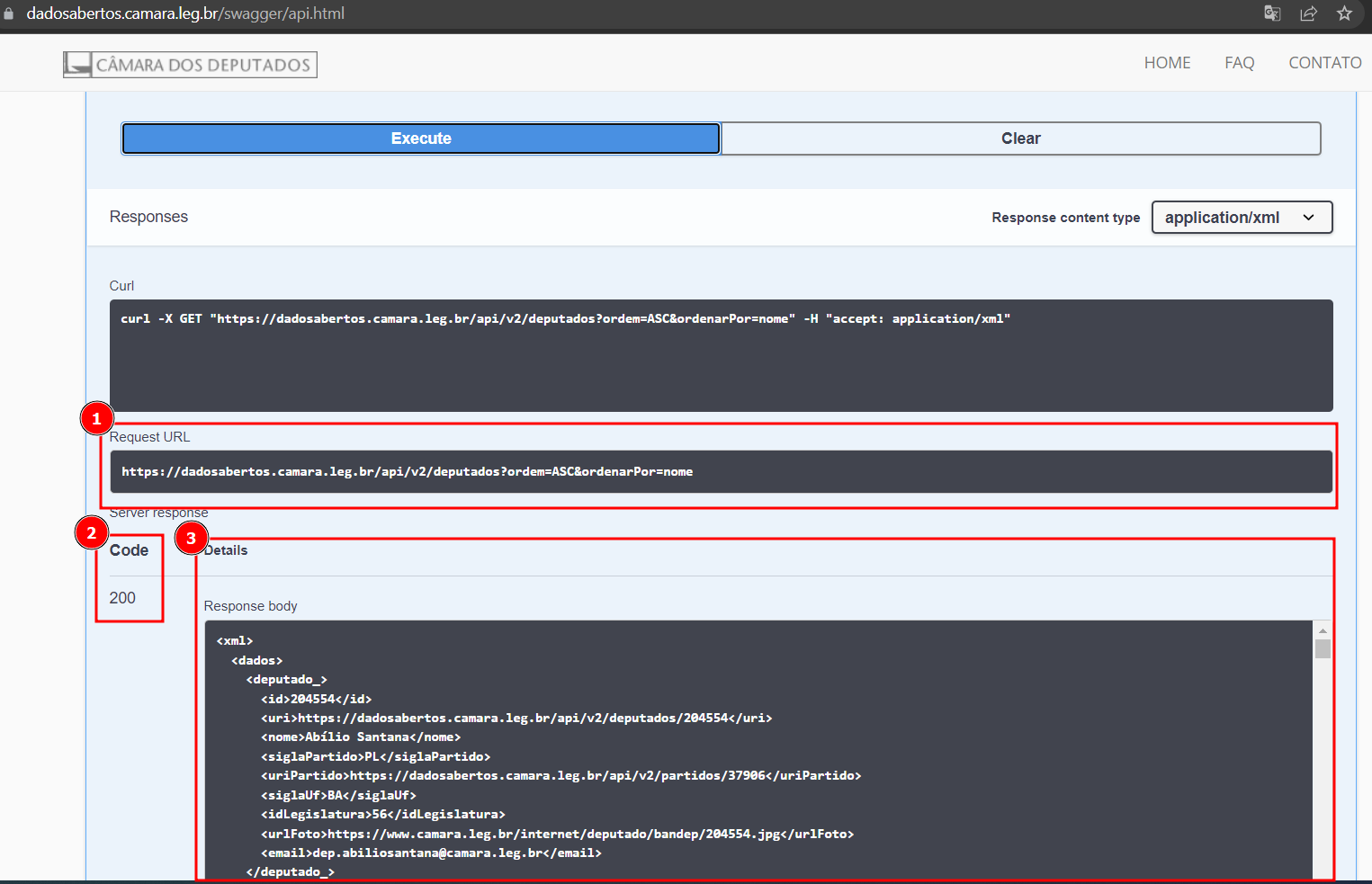
1- Uma breve descrição do que a requisição irá nos retornar

2- Botão para ‘testarmos’ a API e ver o que ela vai nos retornar de informações.

Clicando em Try it out, Rolando a página é possível notar um botão para a execução:



Clicando em execute, o site nos retorna o seguinte:



1 – URL utilizada para fazermos a requisição dos dados

2 – Código que a requisição nos retorna, 200 no caso, significa que a requisição foi bem sucedida.

3 – Os dados que precisamos.

Ótimo, agora que temos a URL para a requisição podemos iniciar nosso código em Python. Nesta primeira etapa do projeto, iremos utilizar apenas duas bibliotecas: Pandas e Requests. Caso não possua elas instaladas, digite em seu terminal:

Pip install pandas

Pip install requests

Caso tenha alguma dificuldade, aqui estão as páginas oficiais

https://pandas.pydata.org/

<https://docs.python-requests.org/en/latest/>

Com as bibliotecas instaladas, Podemos começar:

Vou colocar o código e em seguida explicar o passo a passo de cada linha.

Primeiro vamos importar as bibliotecas que iremos utilizar:

Code

Para fazer as requisições, precisaremos da URL que vimos no passo anterior (item 1).

Code

Agora que temos a URL, Podemos fazer a requisição, para isso usaremos a bblioteca requests importada no inicio do código:

Code

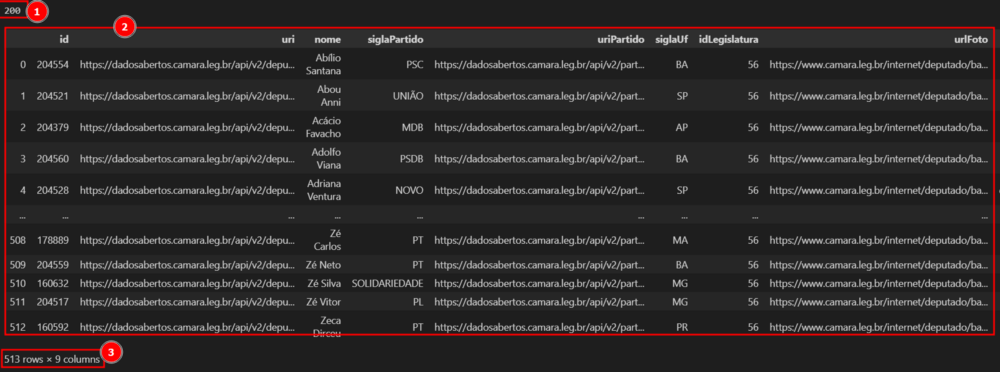
Para sabermos se ocorreu tudo bem iremos mostrar o código de resposta do servidor:

Code

Com tudo certo, utilizaremos a biblioteca do pandas para montarmos um dataframe com os ‘dados’ retornados em formato json:

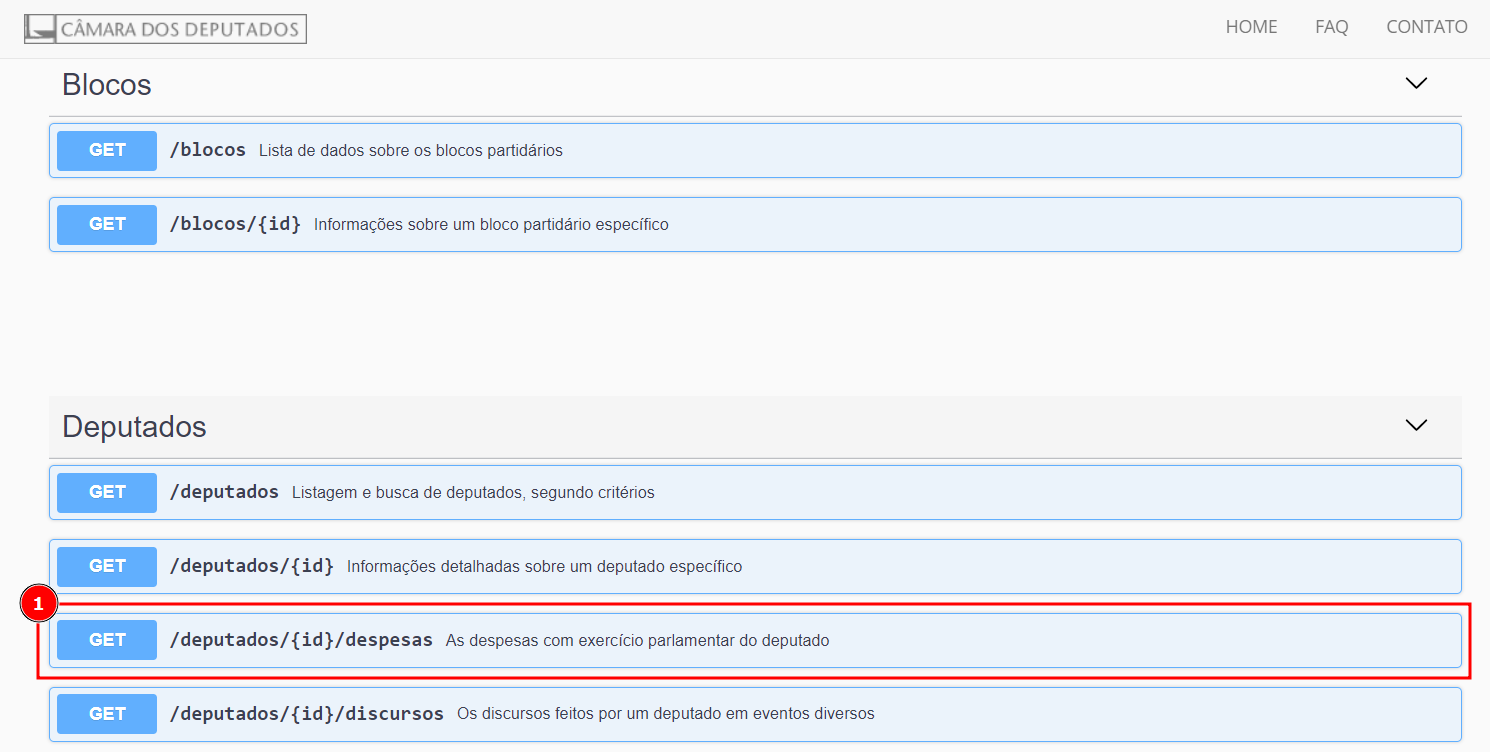
Code

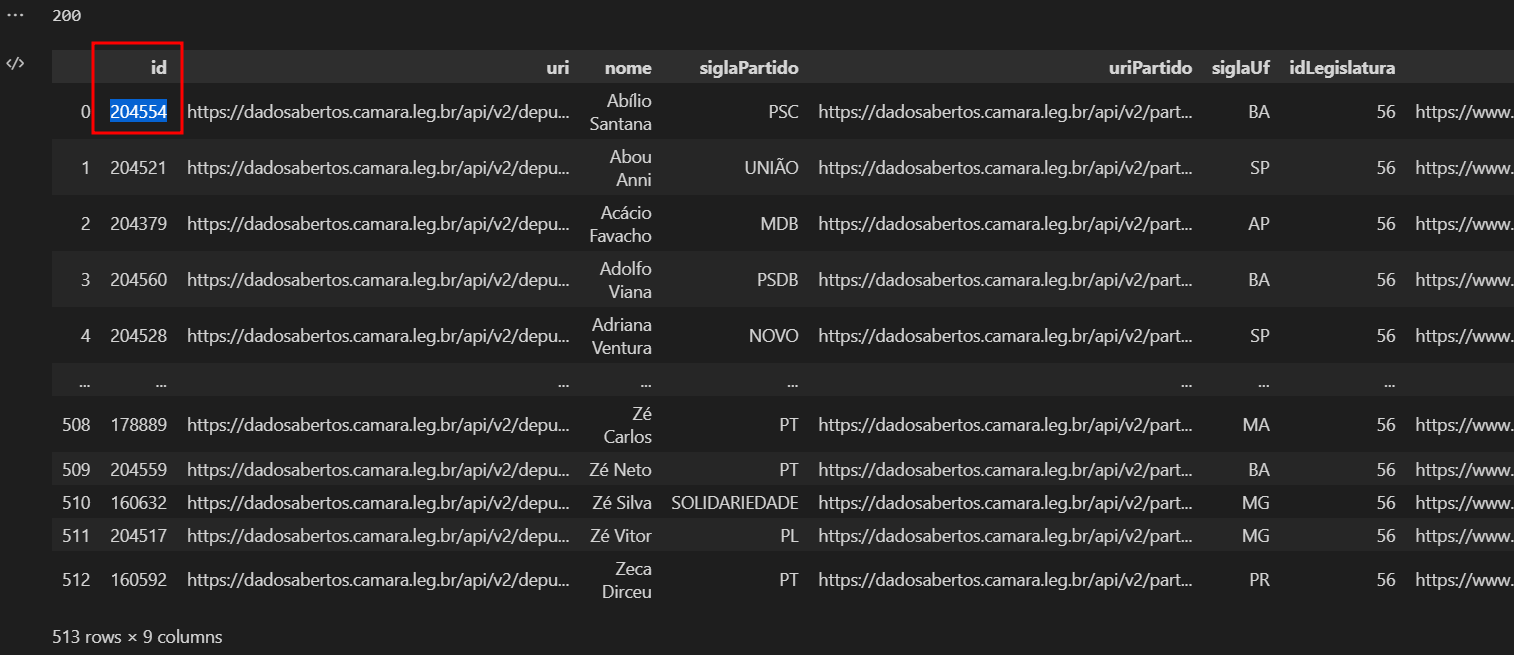
E por fim, nosso resultado:

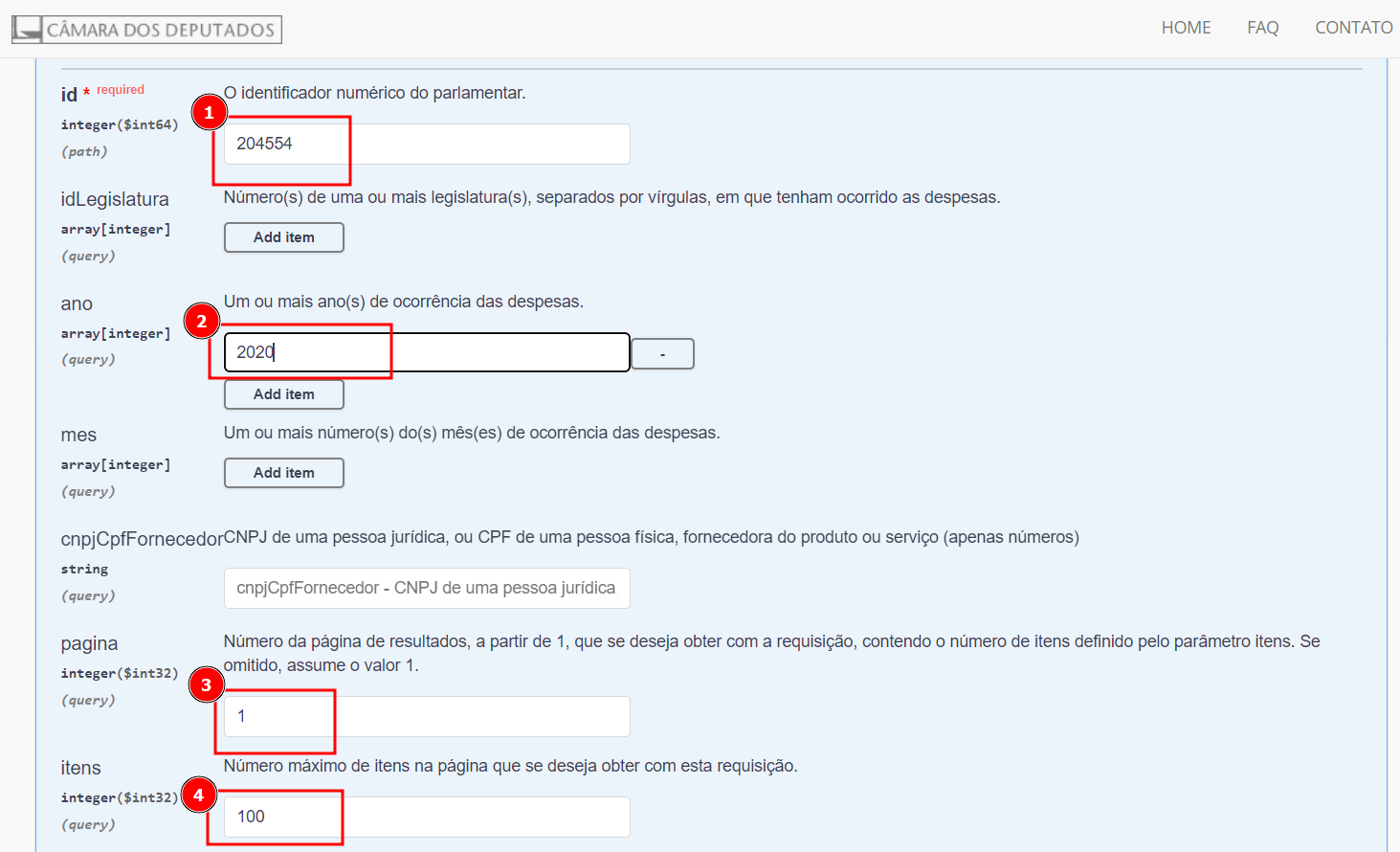


Com essa requisição ja conseguimos informações de todos os deputados, inclusive entre as informações temos o ‘id’ de cada deputado. Com esse ‘id’ conseguiremos fazer as requisições para pegarmos também os dados sobre as despesas.

Voltando na documentação, vamos agora utilizar o primeiro ID que pegamos na requisição anterior para os nossos testes:







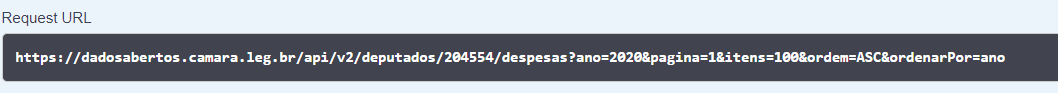
1 – no campo ‘id’ colocamos o primeiro ID que coletamos nan ossa primeira requisição: 204554;

2 – no campo ‘ano’, queremos inicialmente pegar os dados do início da pandemia, no caso: 2020;

3 – no campo ‘página’, queremos a primeira página de despesas então no caso vamos colocar: 1;

4 – no campo ‘itens’, Podemos escolher quantos itens a requisição irá nos retornar por página, Segundo a documentação, se deixarmos o campo em branco por padrão teremos 15 itens por página. Nesse caso iremos escolher retonar 100 itens em cada requisição.

Após ajustar esses parametros clicamos em executar e temos como resultado:



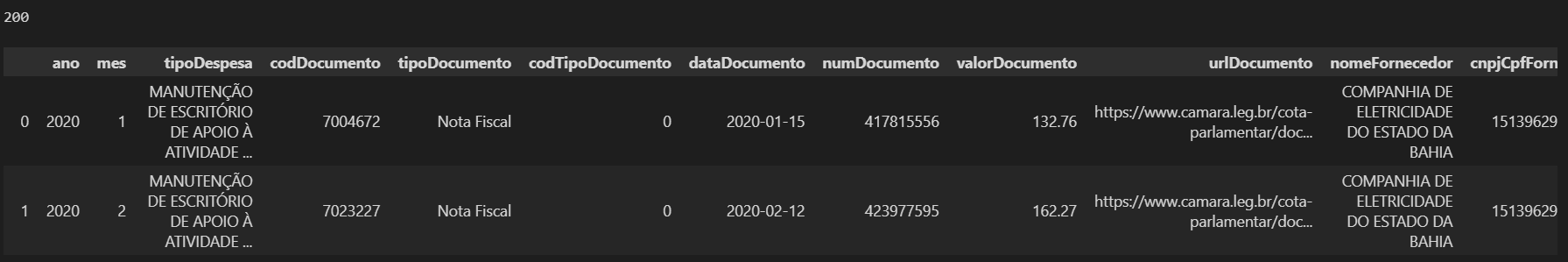
Essa é a URL inicial para fazermos as requisições das despesas:

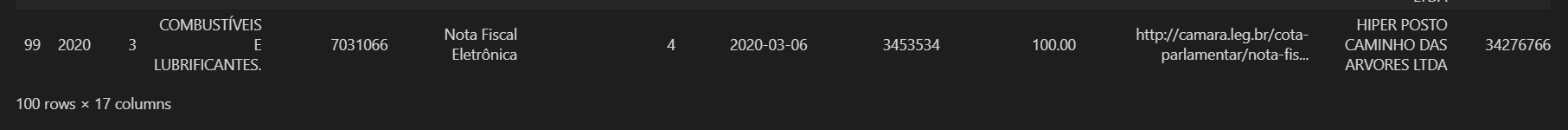
Code

Vamos testar agora utilizando em nosso Código python:

Code

Como resultado temos o seguinte:





Agora temos um dataframe com 100 itens de despesas do deputado com ID: , no ano de 2020. Excelente! Mas e agora, como armazenamos esses dados? Podemos criar uma lista vazia e adicionar os itens do dataframe nela:

Code

E agora temos uma lista com os nossos dados!

Mas como sabemos que nós pegamos todos os dados? Começamos na página 1 mas e se houverem mais páginas, como fazemos para pegar os dados de lá também?

Bom vamos analisar a URL de requisição:



Na URL temos os seguintes parametros que podemos manipular para fazermos as requisições:

1 – ID

2 – Ano

3 – Página

4 – Itens

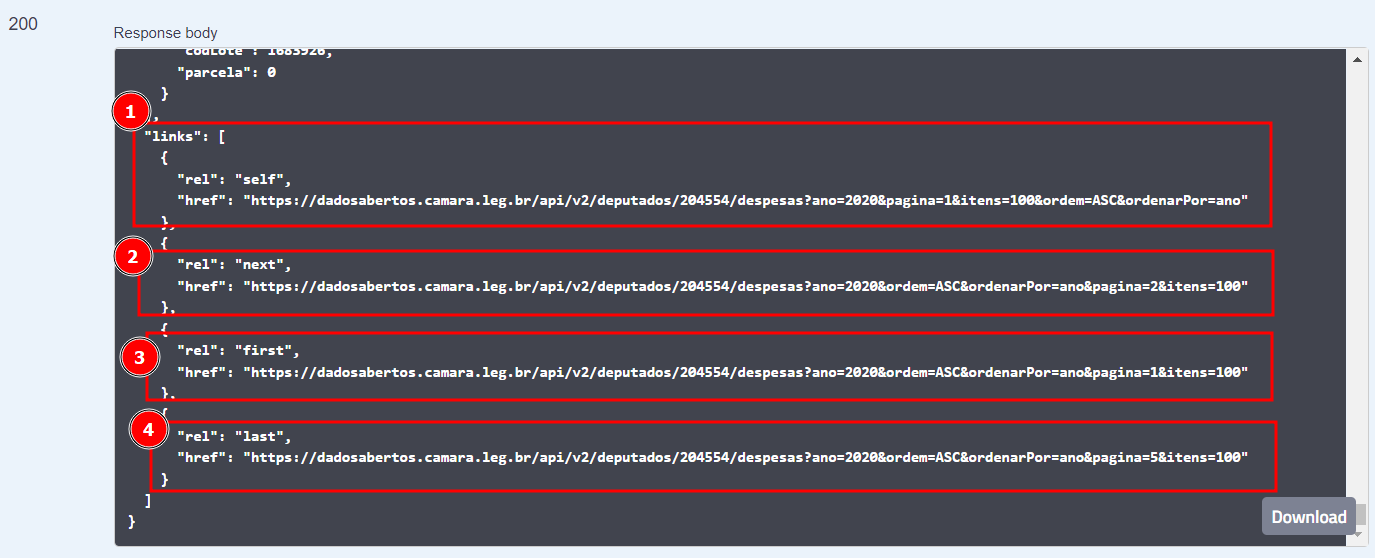
Então se houverem mais páginas, é só mudarmos o parametron ‘pagina’ e fazer outra requisição, mas veremos isso mais adiante.

Como resultado da requisição, nos é retornado o seguinte:



Um dicionário com 100 itens de dados das despesas do deputado de id, no ano de 2020.

Após mostrar os 100 itens, recebemos também os seguintes links:



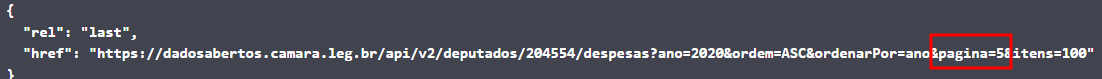
1 – A URL atual

2 – A URL da próxima página (caso houver)

3 – A URL da primeira página

4 – A URL da ultima página.

Esses ultimos links são de grande importancia para nós conseguirmos pegar todas as informações das despesas. Com a URL da ultimá página temos a informação sobre quantas vezes o nosso Código irá se repetir, nesse caso:



Sabemos que para o deputado com Id: , no ano de 2020 teremos 5 páginas de dados para pegar.

Agora que temos essas informações, sabemos que teremos que fazer a paginação das requisições. Faremos primeiro uma função simples que servirá apenas para sabermos qual é a ultima página com informações, tendo essa informação poderemos usar a seguinte lógica:

‘python, vá para a primeira página e pegue o valor da ultima página e chame-a de variável;

Agora pegue as informações de despesas da primeira página e repita esse procedimento alterando o parametron page da URL até o valor de variável’.

Muito confuse?

Vamos ao Código:

Code

Code

Este Código nos retornou os links ja citados anteriormente, nesse caso usaremos apenas o link da ultima página.

Para isolarmos apenas o número da ultima página usaremos expressão regular, caso não tenha essa biblioteca instalada em seu ambiente:

Pip install re

Code

Esse código vai procurar todos os números na string e nos retornar em forma de lista:



Sabemos que a URL é padronizada, então os parametros utilizados estarão sempre na mesma ordem. Neste caso sabemos que o primeiro item da lista faz parte do corpo da URL:

https://dadosabertos.camara.leg.br/api/v2/deputados/

Logo em seguida temos o número de ID do deputado em questão:

204554

Após isso o ano:

2020

O nosso número de interesse no momento que é o número da ultima página:

5

E por fim a quantidade de itens que a página está mostrano:

100

Tendo essas informações podemos criar nossa função que irá precisar de um parametron que no caso será a URL da primeira página, em seguida essa função irá realizar todos os passos que acabamos de fazer e nos retornar apenas o valor que precisamos da ultima página, vamos ao código:

Code