Essa é uma versão dos Guias de Estudo desenvolvida para auxiliar o processo de impressão e para facilitar a leitura dos guias por programas que fornecem a leitura automatizada para suporte a todos os alunos que necessitem. Dessa forma, não apresentaremos ilustrações nesse arquivo.* *#ParaTodosLerem****.*

Módulo #3

Guias de estudo

APIs

Até aqui você já sabe criar seus programas básicos em Python e pode estar querendo aprender como adicionar recursos mais sofisticados nestes programas. Como por exemplo, adicionar integração com outros programas. Como podemos fazer isso?

Suponha que seu objetivo seja desenvolver uma loja online. Nesse processo, você poderá ter a necessidade de gerar boletos bancários como forma de pagamento. Para isso, é necessário utilizar uma API com um banco para gerar estes boletos.

API, nada mais é que um serviço que possibilita a comunicação entre diferentes plataformas e é composta por uma série de padrões e protocolos. Já pensou como é possível o WhatsApp® acessar os contatos que estão em seu celular? Para isso é utilizada uma API que permite a comunicação entre as duas plataformas.

Existem diferentes métodos para uma API, veja:

- *Uma API utiliza o método* ***REST*** *quando quer permitir o acesso remoto a um recurso que o sistema deseja acessar. Esse processo é feito através de requisições HTTP;*
- *Uma API utiliza o método* ***GET*** *quando permitem o acesso remoto para obtenção de dados;*
- *Uma API* ***POST*** *serve para enviar dados;*
- *Já uma API* ***DELETE*** *é utilizado quando o usuário quer apagar algum recurso do servidor.*

Depois que uma requisição é executada, o serviço retorna uma mensagem de resposta HTTP que pode ser uma reposta positiva ou o retorno de um erro. Você já deve ter visto o erro 404 (not found-recurso não encontrado) acontecer quando tenta acessar um site.

Além desses existem outros erros e o que indica cada um deles é o número do status, assim é possível criar mensagens de erro mais claras para o usuário, como: "página não encontrada". Vamos ver os erros mais comuns:

***403** — Acesso Proibido*
***404** — Página/seriviço não encontrado*
***500** — Erro interno do servidor*
***503** — Serviço Indisponível*
***504** — Timeout*

Nada como uma boa prática para entender melhor o conceito de API:

Vamos criar um programa que gere estatísticas sobre a popularidade dos nomes de pessoas. Para isso, vamos utilizar o site https://www.ibge.gov.br/censo2010/apps/nomes/#/search em que para fazer a busca é necessário informar o nome que deseja pesquisar. Por exemplo, se eu buscar pelo nome Milena aparece as seguintes informações:

*Ótimo! Agora quero utilizar estes dados dentro do meu próprio programa. Como podemos fazer? O site disponibilizado abaixo, nos fornece uma **API** que podemos utilizar para fazer a comunicação entre o site e o nosso programa.*

Para acessar os dados sobre o nome Milena precisamos acessar o site da seguinte maneira:

https://servicodados.ibge.gov.br/api/v2/censos/nomes/milena

*Dizemos que o endereço acima e o nosso **endpoint** e ele nos retorna os dados relacionados a quantidade de pessoas registradas com o nome Milena em diferentes décadas:*

[{"nome":"JOAO","sexo":null,"localidade":"BR","res":[{"periodo":"1930[","frequencia":60155},{"periodo":"[1930,1940[","frequencia":141772},{"periodo":"[1940,1950[","frequencia":256001},{"periodo":"[1950,1960[","frequencia":396438},{"periodo":"[1960,1970[","frequencia":429148},{"periodo":"[1970,1980[","frequencia":279975},{"periodo":"[1980,1990[","frequencia":273960},{"periodo":"[1990,2000[","frequencia":352552},{"periodo":"[2000,2010[","frequencia":794118}]}]

Observe bem a estrutura que os dados são armazenados. Você já aprendeu sobre estruturas de dados no curso de Python Iniciante.

Bom, agora vamos então fazer a nossa implementação para recuperar estes dados dentro do nosso programa. Para isso, vamos importar a biblioteca requests e a Json para organizar melhor os dados:

```
import requests
import json
```

Agora, vamos enviar a requisição ao servidor e analisar a resposta que eles retornarão.

```
nome="Milena"
resposta =
requests.get("https://servicodados.ibge.gov.br/api/v2/censos/nomes/Milena"+"home)
resposta

<Response [200]>
```

A resposta contém o **código 200**, que indica que deu certo. Além disso, como a requisição foi bem sucedida, os dados foram enviados na resposta também. Para acessar estes dados vamos fazer seguite:

```
resposta.text
[{"nome":"MILENA","sexo":null,"localidade":"BR","res":[{"periodo":"1930[","frequencia":49},{"periodo":"[1930,1940[","frequencia":80},{"periodo":"[1940,1950[","frequencia":89},{"periodo":"[1950,1960[","frequencia":264},{"periodo":"[1960,1970[","frequencia":887},{"periodo":"[1970,1980[","frequencia":7007},{"periodo":"[1980,1990[","frequencia":18116},{"periodo":"[1990,2000[","frequencia":57995},{"periodo":"[2000,2010[","frequencia":74555}]}]
```

*Para organizar melhor os dados acima, vamos utilizar **Json**, que é uma biblioteca que decodifica um objeto convertendo o formato original para lista e/ou dicionários. Desta maneira, os dados ficam mais fáceis de serem manipulados. Veja:*

```
json_dataset= json.loads(resposta.text)
json_dataset
```

Este código retorna:

```
[{'nome': 'MILENA',
  'sexo': None,
  'localidade': 'BR',
  'res': [{'periodo': '1930[', 'frequencia': 49},
   {'periodo': '[1930,1940[', 'frequencia': 80},
   {'periodo': '[1940,1950[', 'frequencia': 89},
   {'periodo': '[1950,1960[', 'frequencia': 264},
   {'periodo': '[1960,1970[', 'frequencia': 887},
   {'periodo': '[1970,1980[', 'frequencia': 7007},
   {'periodo': '[1980,1990[', 'frequencia': 18116},
   {'periodo': '[1990,2000[', 'frequencia': 57995},
   {'periodo': '[2000,2010[', 'frequencia': 74555}]}]
```

Pronto, agora seu programa já pode utilizar os dados recuperados por meio de uma API.

Lembre-se

Sempre que quiser integrar alguma outra plataforma ao seu programa, analise se a plataforma em questão fornece uma API para facilitar este processo.