

Para saber mais: obtendo arquivos JSON de APIs

Para obter dados no formato JSON podemos utilizar uma **API** (*Application Programming Interface*). Ela é uma interface de programação de aplicação que permite a comunicação entre diferentes programas ou sistemas.

Atenção: Caso queira aprofundar no assunto, temos um [Alura +](https://cursos.alura.com.br/extra/alura-mais/o-que-e-uma-api--c697) (<https://cursos.alura.com.br/extra/alura-mais/o-que-e-uma-api--c697>) explicando o que é uma API.

Existem vários formatos que podem ser usados em APIs e um dos mais comuns é o **JSON** (*JavaScript Object Notation*). O JSON é um formato leve de troca de informações entre sistemas que utiliza uma notação baseada em objetos do **JavaScript**.

Nós temos a seguinte situação-problema:

“Uma empresa de alimentos saudáveis está enfrentando um problema com a popularidade de seus produtos. Apesar de oferecer opções saudáveis, muitos(as) clientes estão relutantes em comprar os produtos devido à falta de informação sobre os valores nutricionais das frutas que são utilizadas na produção dos alimentos”.

Nesta situação podemos acessar uma API chamada [Fruitvice](https://fruityvice.com/) (<https://fruityvice.com/>) para coletar informações detalhadas sobre os valores nutricionais das frutas que são usadas nos produtos. Com essas informações, a empresa pode atualizar suas embalagens e materiais de marketing para destacar os valores nutricionais das frutas e, assim, atrair mais clientes que buscam opções saudáveis. Além disso, a empresa pode utilizar as informações para desenvolver novos produtos que atendam ainda mais às necessidades nutricionais de sua clientela.

Uma maneira comum de acessar APIs é por meio da biblioteca [requests](https://pypi.org/project/requests/) (<https://pypi.org/project/requests/>). Para isso, também é necessário importar o módulo `json` :

```
import requests
import json
```

[COPIAR CÓDIGO](#)

Os dados da **API Fruitvice** serão obtidos com a função `request.get('url')`. Dentro dela nós passamos o endereço da API seguido de `api/fruit/all` para obter os dados de todas as frutas.

```
dados_frutas = requests.get('https://fruityvice.com/api/fruit/all')
```

[COPIAR CÓDIGO](#)

Conseguimos recuperar os resultados usando a função `json.loads()`. Para isso, é necessário passar a variável `dados_frutas`, criada anteriormente com o atributo `text` que retorna o conteúdo da resposta.

```
resultado = json.loads(dados_frutas.text)
```

[COPIAR CÓDIGO](#)

Em seguida, podemos visualizar o DataFrame.

```
pd.DataFrame(resultado)
```

[COPIAR CÓDIGO](#)

index	genus	name	id	family	order	nutritions
0	Malus	Apple	6	Rosaceae	Rosales	{'carbohydrates': 11.4, 'protein': 0.3, 'fat': 0.4, 'calories': 52}
1	Prunus	Apricot	35	Rosaceae	Rosales	{'carbohydrates': 3.9, 'protein': 0.5, 'fat': 0.1, 'calories': 15}
2	Persea	Avocado	84	Lauraceae	Laurales	{'carbohydrates': 8.53, 'protein': 2, 'fat': 14.66, 'calories': 160}
3	Musa	Banana	1	Musaceae	Zingiberales	{'carbohydrates': 22, 'protein': 1, 'fat': 0.2, 'calories': 105}
4	Rubus	Blackberry	64	Rosaceae	Rosales	{'carbohydrates': 9, 'protein': 1.3, 'fat': 0.4, 'calories': 63}
5	Fragaria	Blueberry	33	Rosaceae	Rosales	{'carbohydrates': 5.5, 'protein': 0, 'fat': 0.4, 'calories': 57}
6	Prunus	Cherry	9	Rosaceae	Rosales	{'carbohydrates': 12, 'protein': 1, 'fat': 0.3, 'calories': 63}

Agora sabemos como utilizar a biblioteca requests e o módulo JSON para obter dados de APIs!