RPA na PRÁTICA

Aula – APIs

O que é uma API?

Uma forma bem prática para poder entender isso, é imaginar um restaurante. Nele existe o cliente, o garçom e o cozinheiro.

1. cliente chega, senta e chama o garçom;
2. garçom vem até a mesa, anota o pedido e leva ao chefe de cozinha;
3. chefe de cozinha prepara o prato (processa o pedido);
4. garçom coleta o prato pronto e leva para o cliente que pediu;
5. cliente final consome o prato preparado.

A API seria basicamente isso, um serviço que leva um dado, processa e devolve o resultado. As páginas webs são exemplos de serviços, onde o navegador envia o pedido e o servidor devolve o código da página para que seja renderizado como imagem ao usuário.

Quais os tipos de APIs?

As APIs possuem duas modalides, estas sendo REST ou SOAP.
A tecnologia REST é mais recente e está sendo mais utilizada pelo mercado, já a SOAP foi muito utilizada por programas mais antigos (legados).

Como podemos testar uma API?

Programas

REST

- Postman (https://www.postman.com/downloads/)
- Insomnia (https://insomnia.rest/download)

SOAP

- SoapUI (https://www.soapui.org/downloads/soapui/)

Código

requests

- Instalando a "lib" requests

```
pip install requests
```

http

- Importando a "lib" http.client

```
from http.client import HTTPSConnection
```

urllib

- Importando a "lib" urllib.request

```
from urllib.request import urlopen
```

GET

Explicação

As APIs do tipo GET são de modelagem simples, isso é, a requisição é feita somente tendo uma URL e nela podem ser passado parâmetros extras para serem efetuados filtros e ajustes no retorno devolvido por ela.
Obs.: Devido a mesma ter uma facilidade na chamada, o tipo GET pode se tornar perigoso se for necessário passar credenciais ou dados sigilosos.

Exemplo

1. Importando a "lib" requests

```
import requests
```

2. Efetuando requisição

```
url = "http://universities.hipolabs.com/search?country=brazil"


response = requests.get(url)
```

3. Transformando o retorno em dicionário

```
r_json = response.json()
```

4. Interagindo com o retorno

```
print(r_json[15]['name'])
# output: 'Universidade Regional de Blumenau'
```

APIs Para Praticar

- ViaCEP (https://viacep.com.br/)
- JssonPlaceholder (https://jsonplaceholder.typicode.com/)
- Banco Central (https://www.bcb.gov.br/api/servico/sitebcb/indicadorinflacao)
- Agify (https://agify.io/)
- University Domains (https://github.com/Hipo/university-domains-list)

POST

Explicação

O tipo POST permite a passagem de dados/parâmetros através de um "corpo" que fica "mascarado" durante o envio e o retorno. Logo, ela tem uma base mais complexa para ser utilizada, podendo até mesmo ter autenticações baseadas em Token.
Obs.: Os "corpos" geralmente são no formato JSON.

Exemplo

1. Importando a "lib" requests e JSON

```
import requests
import json
```

2. Gerando "corpo" da API

```
payload = json.dumps({
  "name": "Harry Potter",
  "job": "Desenvolvedor RPA"
})
```

3. Gerando os cabeçalhos da API

```
headers = {
  'Content-Type': 'application/json'
}
```

4. Efetuando requisição

```
url = "https://reqres.in/api/users"


response = requests.post(url, headers=headers, data=payload)
```

5. Transformando o retorno em dicionário

```
r_json = response.json()
```

6. Interagindo com o retorno

```
print(r_json.get('createdAt'))
# output: '2022-05-23T19:53:22.594Z'
```

RPA na PRÁTICA