Python e Webscraping

Aprendendo a extrair dados da Web

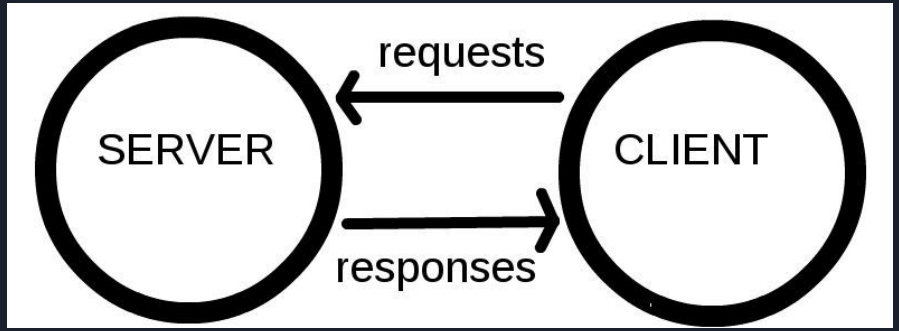
Aula 2 - Webscraping

Felipe Andrade
felipe@potelo.com.br

O que é Webscraping?

- Técnica para extrair dados de websites;
- Scraping == raspagem.

Como funciona a web?

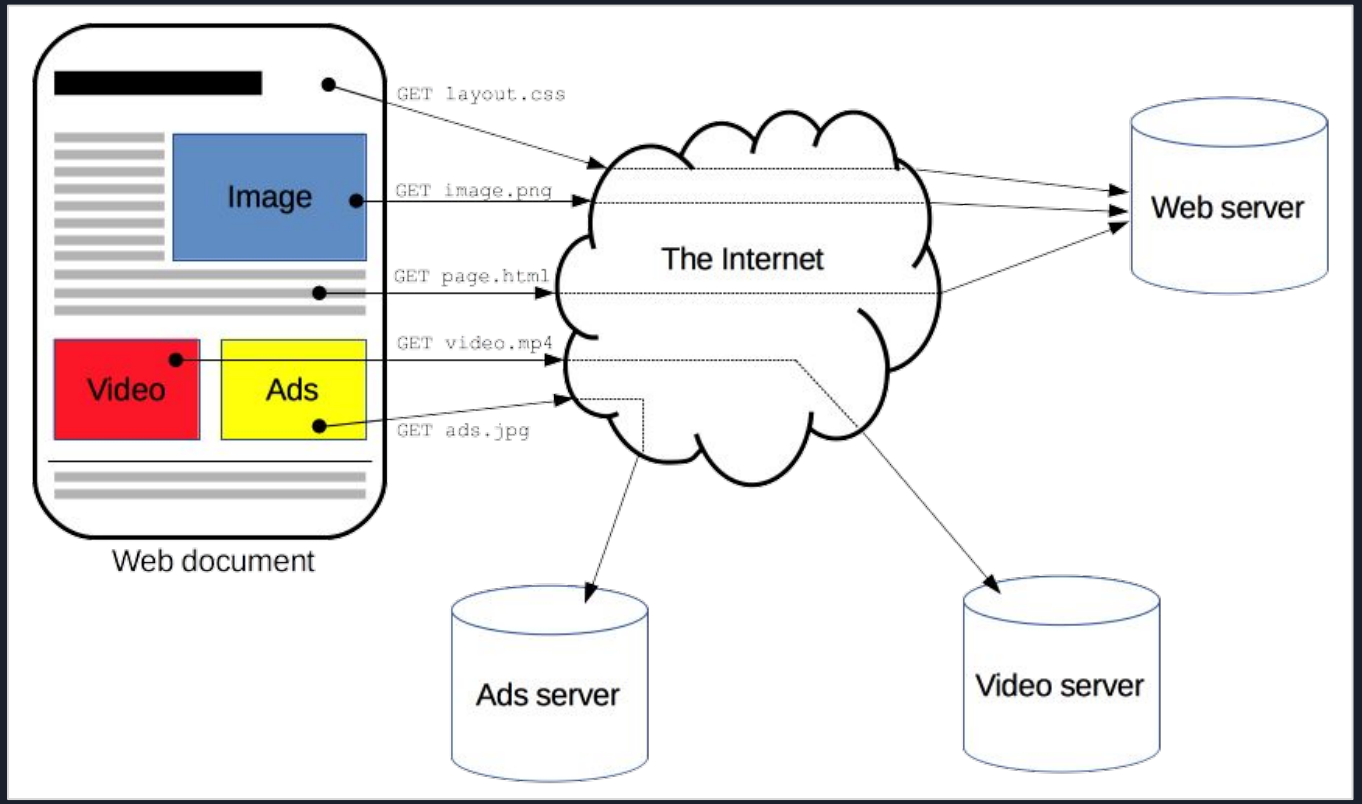


- Client = usuários da web conectados à internet;
- Server = computadores que armazenam as páginas ou aplicativos.

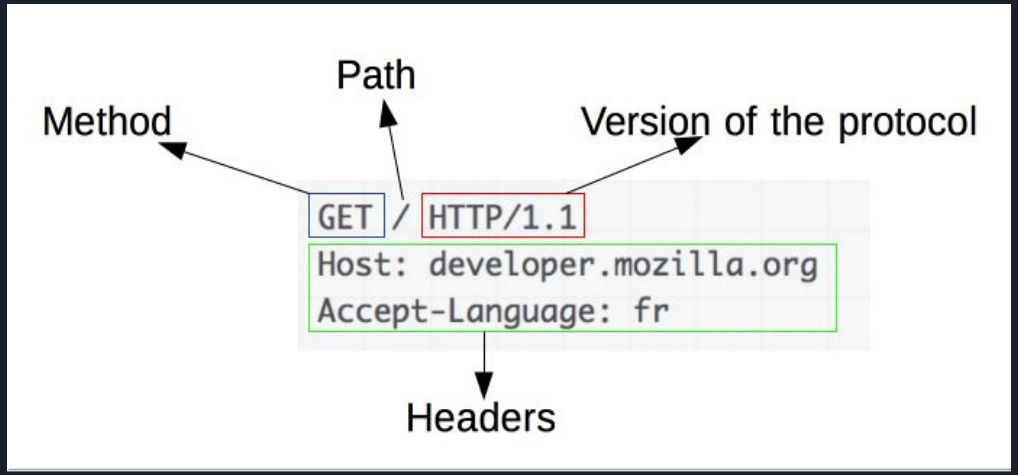
Protocolo HTTP

- Hypertext Transfer Protocol ou Protocolo de Transferência de Hipertexto ;
- É um protocolo cliente-servidor;
- O cliente envia uma requisição(*request*) e o servidor devolve uma resposta(*response*);

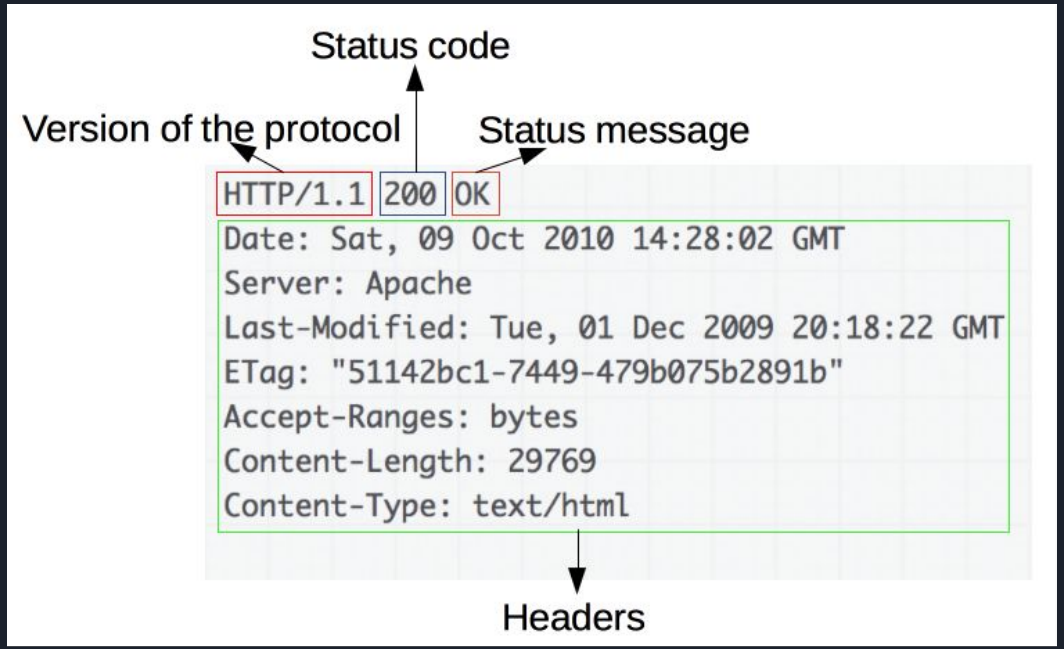
HTTP


GET layout.css
GET image.png
GET page.html
GET video.mp4
GET ads.jpg
Image
Video
Ads
Web document
The Internet
Web server
Ads server
Video server

HTTP - Request



HTTP - Response



HTTP - Métodos

- GET: requisita um recurso especificado;
- POST: envia dados para serem processados;
- PUT: edita um recurso;
- DELETE: exclui um recurso;

Veja todos os metodos em: https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html

HTTP - Códigos de retorno

- 2xx: Sucesso – indica que a requisição foi bem sucedida. Ex: 200 - OK
- 3xx: Redirecionamento – informa que uma nova requisição será feita internamente. Ex.: 302 - Encontrado.
- 4xx: Erro no cliente – avisa que o cliente fez uma requisição que não pode ser atendida. Ex.: 404 - Não encontrado
- 5xx: Erro no servidor – ocorreu um erro no servidor ao cumprir uma requisição válida. Ex.: 500 - Erro interno do servidor.

Veja todos os códigos em: https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

Requisição HTTP com Python

Veja a documentação em: http://docs.python-requests.org/

Exemplo de uma requisição com Requests

```
>>> import requests
>>> response = requests.get("http://www.correio24horas.com.br/noticias/")
>>> response.status_code
200
>>> response.headers
{'Date': 'Sat, 09 Dec 2017 06:16:13 GMT', 'Content-Type': 'text/html; charset=utf-8', 'Vary': 'Accept-Encoding', 'Content-
Language': 'pt_BR', 'Content-Encoding': 'gzip', 'X-App': 'correio_node3.correio_portal', 'Cache-Control': 'no-cache, must-
revalidate', 'Pragma': 'no-cache', 'Age': '0', 'X-Cache': 'MISS', 'X-Device': 'pc', 'X-Origin': 'web', 'Grace': 'none', 'A
ccept-Ranges': 'bytes', 'Transfer-Encoding': 'chunked', 'Connection': 'keep-alive', 'Set-Cookie': 'LWF5=2726873786.20480.0
000; path=/'}
>>> with open('noticias.html', 'w') as f:
...     f.write(response.text)
...
```

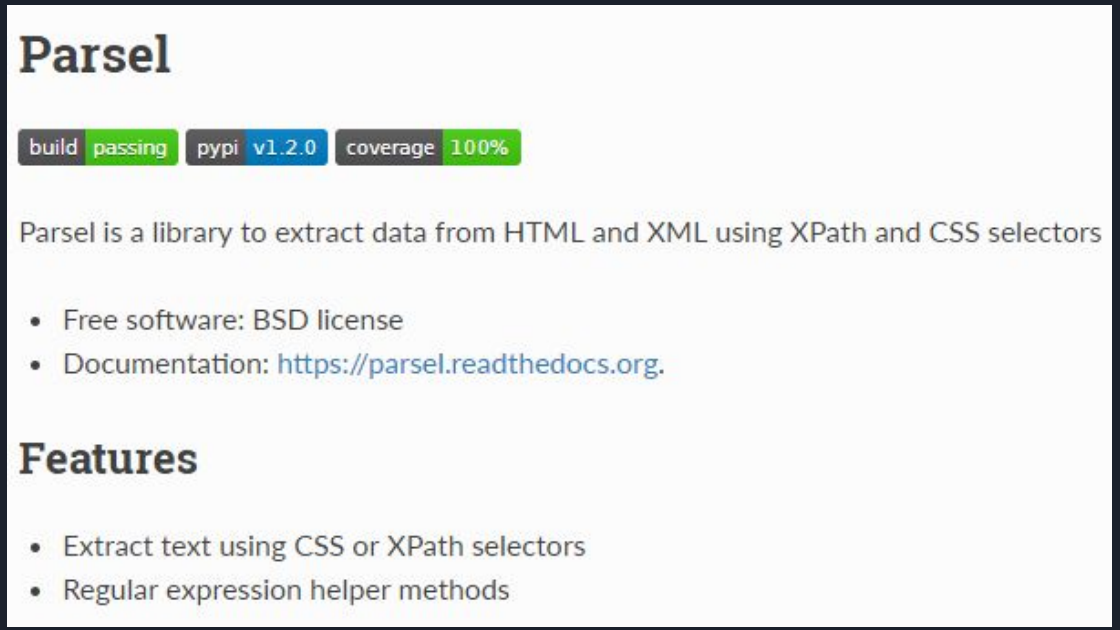
Como fazer uma requisição POST?

Exemplo de um POST com Requests

```
payload = {
        'email': felipe@eu.com',
        'password': 123456
}

requests.post("https://www.packtpub.com/register", data=payload,
headers=headers)
```

Extraindo dados de um HTML



Parsel

build passing pypi v1.2.0 coverage 100%

Parsel is a library to extract data from HTML and XML using XPath and CSS selectors

- Free software: BSD license
- Documentation: https://parsel.readthedocs.org.

Features

- Extract text using CSS or XPath selectors
- Regular expression helper methods

Veja a documentação em: http://parsel.readthedocs.io

Extraindo dados de um HTML

```
>>> import requests
>>> from parsel import Selector
>>> response = requests.get("http://ba.olx.com.br")
>>> response.status_code
200
>>> titulos = Selector(response.text).xpath(".//ul[@class='list']//a[@class='OLXad
-list-link']/@title").extract_first()
>>> print(titulos)
Cabo iphone
```

Hora de codar!

Felipe Andrade
felipe@potelo.com.br