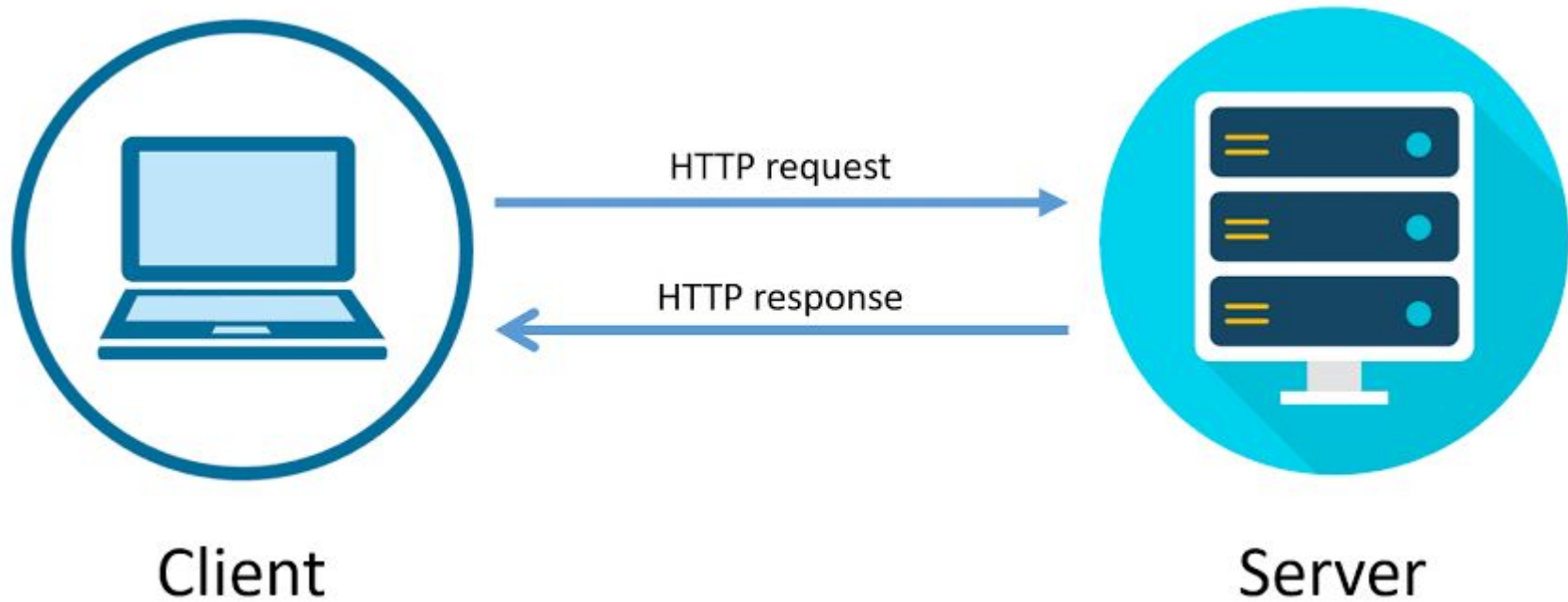


Fundamentos e Práticas com Python para Engenharia de Dados

Faber Xavier - Aula 2

Request



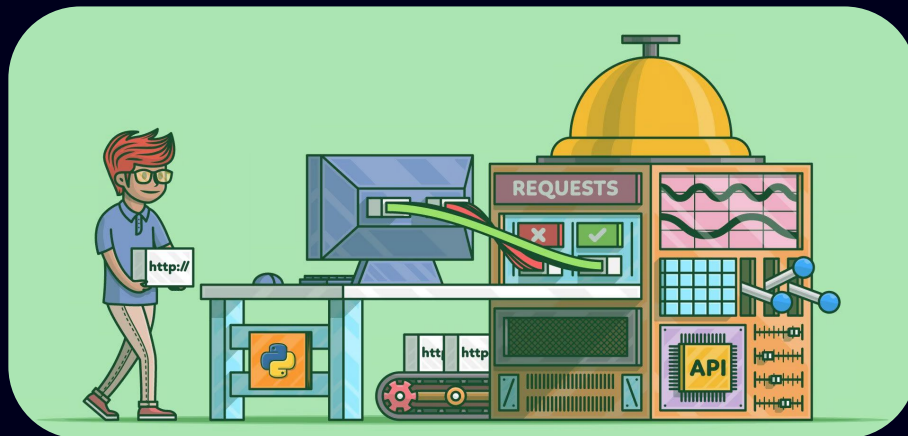
Request

- Biblioteca HTTP para Python
 - Alternativa a libs nativas urllib
- Envia requisições HTTP/1.1 com facilidade
- Interface simples e intuitiva
- <https://pypi.org/project/requests/>

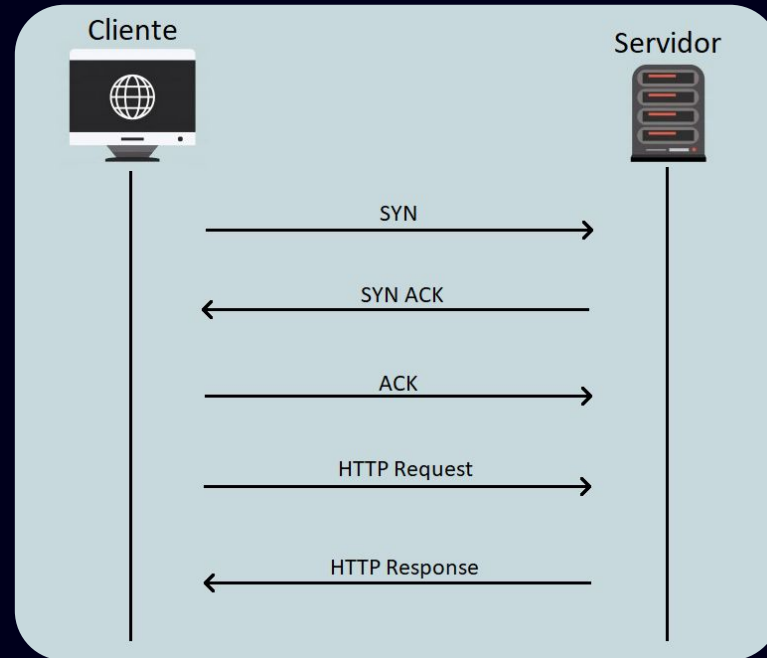
```
>>> import requests
>>> r = requests.get('https://httpbin.org/basic-auth/user/pass', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"authenticated": true, ...}'
>>> r.json()
{'authenticated': True, ...}
```

Request

- Sintaxe simples
- Extensa documentação
- Suporte a métodos HTTP
- Controle avançado (cookies, headers, timeout)



Request



Request

- Primeira requisição HTTP

```
import requests

response = requests.get('https://api.github.com')
print(response.status_code)
```

Retorno:

- response.status_code
- response.headers
- response.text
- response.json()

Request

Item	Finalidade	Tipo de Retorno
<code>response.status_code</code>	Código HTTP da resposta	Inteiro (int)
<code>response.headers</code>	Cabeçalhos da resposta	Dicionário-like
<code>response.text</code>	Corpo da resposta como texto	String (str)
<code>response.json()</code>	Corpo da resposta como objeto Python (JSON parse)	Dicionário ou lista (dict ou list)



Request

- **Payload**
 - Payload é o dado ou conteúdo que você está enviando em uma requisição HTTP.
- O payload é o “conteúdo útil”: os dados de formulário, os arquivos enviados, um objeto JSON, etc.

```
import requests

payload = {'username': 'fabio', 'password': '1234'}
response = requests.post('https://example.com/login', data=payload)
```

```
payload = {'search': 'python requests'}
response = requests.post('https://example.com/api/search', json=payload)
```


Request

- **Post**

- Para enviar dados em um formulário, usamos o método post. O dicionário passado para o data será enviado no corpo da requisição

```
payload = {'username': 'user', 'password': 'pass'}  
r = requests.post('https://httpbin.org/post', data=payload)
```

```
import json  
  
payload = {'key': 'value'}  
r = requests.post('https://httpbin.org/post', json=payload)
```

Request

- **Headers**

- Algumas APIs exigem cabeçalhos específicos, como autenticação ou identificação de aplicativo. Podemos enviar headers personalizados usando o argumento headers.

```
headers = {'User-Agent': 'my-app/0.0.1'}  
r = requests.get('https://api.github.com', headers=headers)
```

Request

- **Timeout**

- Para evitar que a aplicação trave indefinidamente esperando uma resposta, usamos o timeout. É possível também implementar tentativas automáticas (retries) com bibliotecas auxiliares como urllib3.

```
r = requests.get('https://api.github.com', timeout=5)
```

Obrigado

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

Please keep this slide for attribution



PUC Minas