



REST API

Live de Python # 175



1. API / REST

Como é uma API? O que é REST?

2. Verbos

Como dizer o que eu quero?

3. Respostas

Entendendo os códigos de resposta

4. Filtros

Como especificar o que eu quero?



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto



Ademar Peixoto, Alex Lima, Alex Lopes, Alexandre Harano, Alexandre Santos, Alexandre Tsuno, Alexandre Villares, Alynne Ferreira, Alysson Oliveira, Amaziles Carvalho, André Rocha, Arnaldo Turque, Artur Zalewska, Bruno Batista, Bruno Oliveira, Caio Nascimento, Carlos Chiarelli, Cleber Santos, César Almeida, Davi Ramos, David Kwast, Diego Guimarães, Diego Ubirajara, Dilenon Delfino, Donivaldo Sarzi, Elias Soares, Eric Niens, Eugenio Mazzini, Everton Alves, Fabiano Gomes, Fabio Barros, Fabio Castro, Fabrícia Diniz, Flavkaze Flavkaze, Francisco Alencar, Franklin Silva, Fábio Serrão, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Castro, Guilherme Felitti, Guilherme Ostrock, Henrique Machado, Israel Fabiano, Italo Silva, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jorge Plautz, Jose Mazolini, José Prado, João Lugão, João Schiavon, Juan Gutierrez, Jônatas Silva, Júlia Kastrup, Kaio Peixoto, Kaneson Alves, Leonardo Cruz, Leonardo Galani, Leonardo Mello, Lidiane Monteiro, Lorena Ribeiro, Lucas Barros, Lucas Mello, Lucas Mendes, Lucas Teixeira, Lucas Valino, Luciano Ratamero, Maiquel Leonel, Marcela Campos, Marcelo Rodrigues, Maria Clara, Marina Passos, Matheus Vian, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Patric Lacouth, Patricia Minamizawa, Patrick Gomes, Paulo Tadei, Pedro Pereira, Peterson Santos, Rafael Lino, Reinaldo Silva, Renan Moura, Revton Silva, Rodrigo Ferreira, Rodrigo Mende, Rodrigo Vaccari, Ronaldo Silva, Sandro Mio, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Tyrone Damasceno, Vinícius Bastos, Vítor Gomes, Wendel Rios, Wesley Mendes, Willian Lopes, Willian Rosa, Wilson Duarte, Yury Barros, Érico Andrei



Obrigado você



Afinal o que é
uma API?

API

O que é uma API?

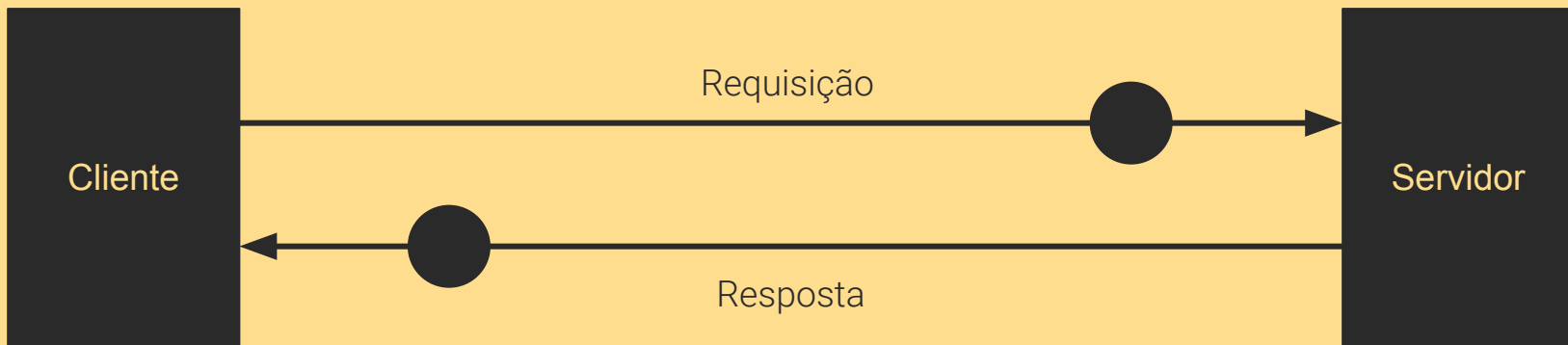


Uma API (Application Programming Interface) é um conjunto de definições e protocolos usados no desenvolvimento e na integração de aplicações.

O que é uma API?



Uma API é um conjunto de definições e protocolos usados no desenvolvimento e na integração de aplicações.



Um contrato

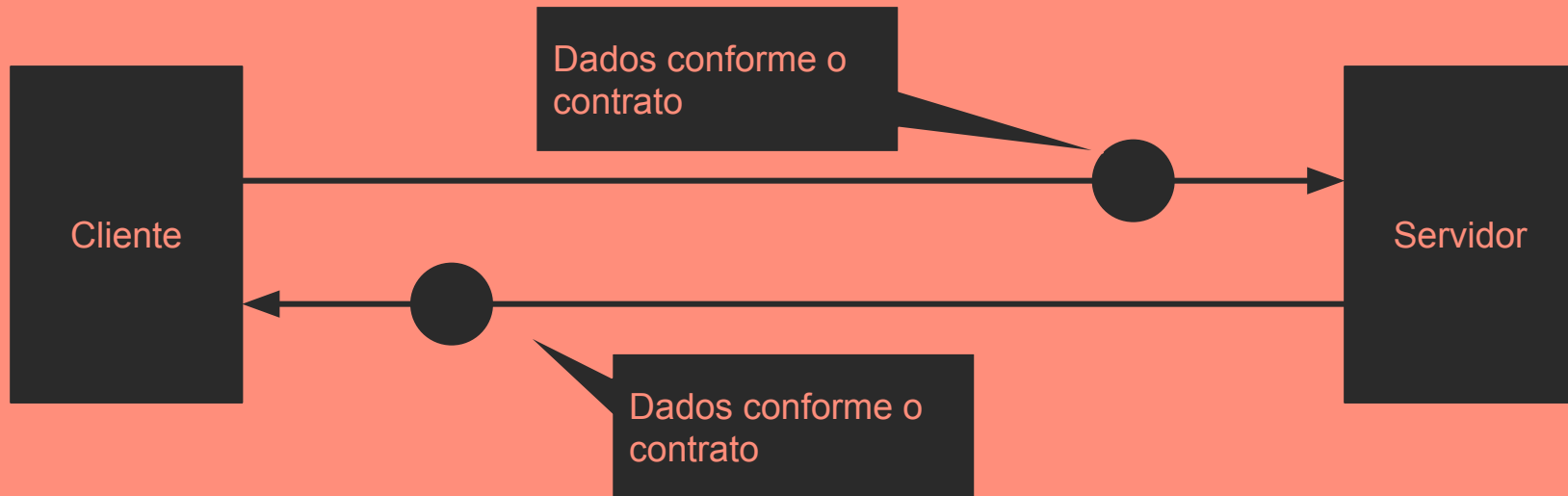


Basicamente uma api é **contrato** de como os dados serão enviados e quais respostas vamos obter quando um dado específico for solicitado.

Um contrato



Basicamente uma API é **contrato** de como os dados serão enviados e quais respostas vamos obter quando um dado específico for solicitado.



Outra sigla difícil?

REST

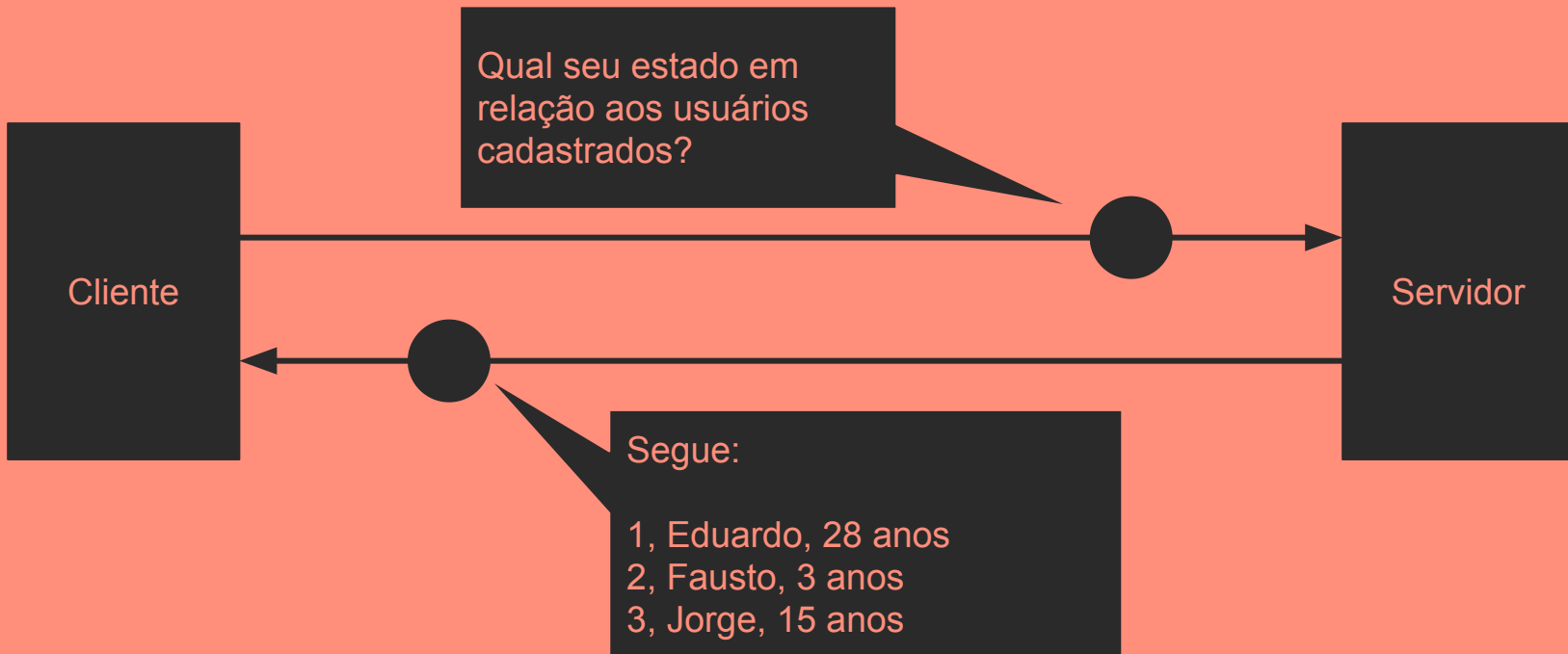
O que é REST?



REST (Representational State Transfer) é um conjunto de "regras" e limitações para que a API não seja complexa demais.

Basicamente o que pedimos a uma API é o seu **estado** [State] e ela nos retorna uma **Representação** [Representational] dele em uma **Transferência** [Transfer] de dados.

O que é REST?



Que ferramentas podemos usar em Python?



Flask



Falcon



CherryPy



AIOHTTP



Tornado

Que ferramentas podemos usar em Python?



Flask



Falcon



CherryPy



AIOHTTP



Tornado

```
pip install flask
```



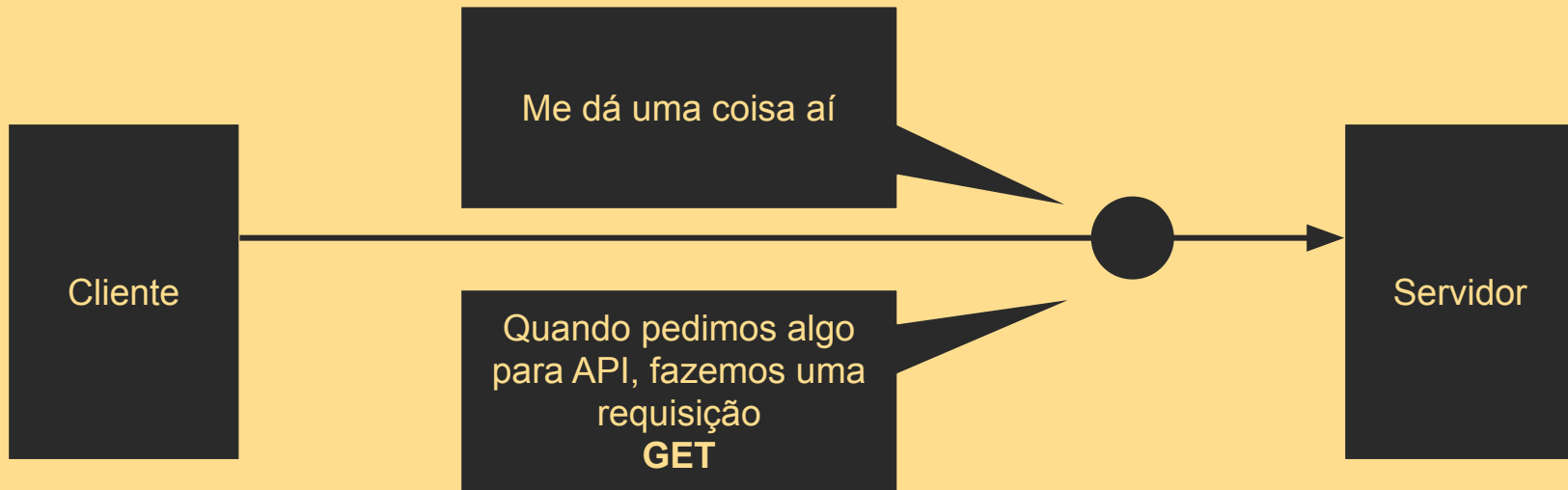
Vamos instalar o Flask



Verbos

Como dizer o que
eu quero?

Como funciona a comunicação?



```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.get('/recurso')
6  def respondo_estado():
7      return 'OK'
8
9  app.run( )
```



Nossa primeiro exemplo



```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.get('/recurso')
6  def respondo_estado():
7      return 'OK'
8
9  app.run( )
```



Nossa primeiro exemplo



Tá, OK! Mas como rodar esse código?



```
1 flask run
```

```
pip install httpie
```



Instalando um cliente



Como conversar com ele?



Servidor

Nosso servidor, tem um endereço.
(http://localhost)

Tem uma porta (5000)

E precisamos acessar um recurso ou como costumamos chamar, endpoint

http://localhost:5000/recurso



```
http get http://localhost:5000
```



0 comando com httpie



A especificação

Cont
rato

Contrato



Lembra que falamos antes sobre contrato? Existe um formato padrão para especificação que é o **OpenAPI**. [Esse tema vale uma live???]

Que é como dizemos o que deve ser enviado API e o que será respondido quando um pedido for feito

```
pip install flask-pydantic-spec
```



OpenAPI



Sensohive API

New Tab

← → ↺

http://petstore.swagger.io/?url=https://sensohive.com/docs/api/swagger.json

★ ⋮

swagger

https://sensohive.com/docs/api/swagger.json

Explore

Sensohive API ^{1.3.1}

[Base URL: `api.sensohive.com/v1`]
<https://sensohive.com/docs/api/swagger.json>

This is the specification of the Sensohive API
[Contact the developer](#)
[README about the API](#)

Schemes

HTTPS

Authorize

device

Device information

Find out more: <https://sensohive.com/docs/api/>

GET	/dev	List all your devices	🔒
GET	/dev/{id}	Find a device	🔒
GET	/dev/{id}/data	Get data	🔒
PUT	/dev/{id}/user/{username}	Set device user	🔒
DELETE	/dev/{id}/user	Revoke user access	🔒
PUT	/dev/{id}/webhooks/{webhookname}	Add device webhook	🔒
DELETE	/dev/{id}/webhooks/{webhookname}	Relinquish device webhook	🔒
GET	/dev/{id}/info	Get device information	🔒

user

User management

The user of a device: <https://sensohive.com/docs/api/>

GET	/users	List all your users	🔒
-----	--------	---------------------	---

Como esses
dados devem ser
representados?

Represe
ntação

Representação



Existem vários formatos para a representação desses dados. O formato mais comum atualmente é **json**.

Embora a representação do dado possa ser feita usando **XML**, e também **YAML**

Representação



```
{  
    "field": "value",  
    "chave": "valor",  
    "nome": "Eduardo"  
}
```

Representação



```
1  from pydantic import BaseModel
2
3
4  class Pessoa(BaseModel):
5      id: Optional[int]
6      nome: str
7      idade: int
```

Voltando aos Verbos



O protocolo HTTP é a base usada por trás das APIs REST e as “chamadas” são feitas partindo de “tipos” de requisições. Vamos conversar sobre as mais populares, mas ao total são 9 verbos.

GET	Pedir os recursos, dados. Ler o que está lá no servidor
POST	Criar um recurso, inserir um dado / registro
PUT	Alterar um registro (mandando todos os campos)
DELETE	Deletar um recurso / registro / dado

Voltando aos Verbos



O protocolo HTTP é a base usada por trás das APIs REST e as “chamadas” são feitas partindo de “tipos” de requisições. Vamos conversar sobre as mais populares, mas ao total são 9 verbos.

GET	Seguro / Indepotente	Pedir os recursos, dados. Ler o que está lá no servidor
POST		Criar um recurso, inserir um dado / registro
PUT	Indepotente	Alterar um registro (mandando todos os campos)
DELETE	Indepotente	Deletar um recurso / registro / dado

Respo stas

Entendendo as
respostas

CÓDIGOS



Para cada requisição no servidor, independentemente do verbo de “chamada” retorna alguns códigos de resposta. Eles são divididos em 5 grupos. Vamos conversar só sobre os mais recorrentes

200	Códigos de sucesso
201	Criado
204	Sem conteúdo (usado para deletar com sucesso, por exemplo)
400	Códigos de erro
404	Não encontrado
500	Erro no servidor

Filtros

Melhorando
nossa busca



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto

