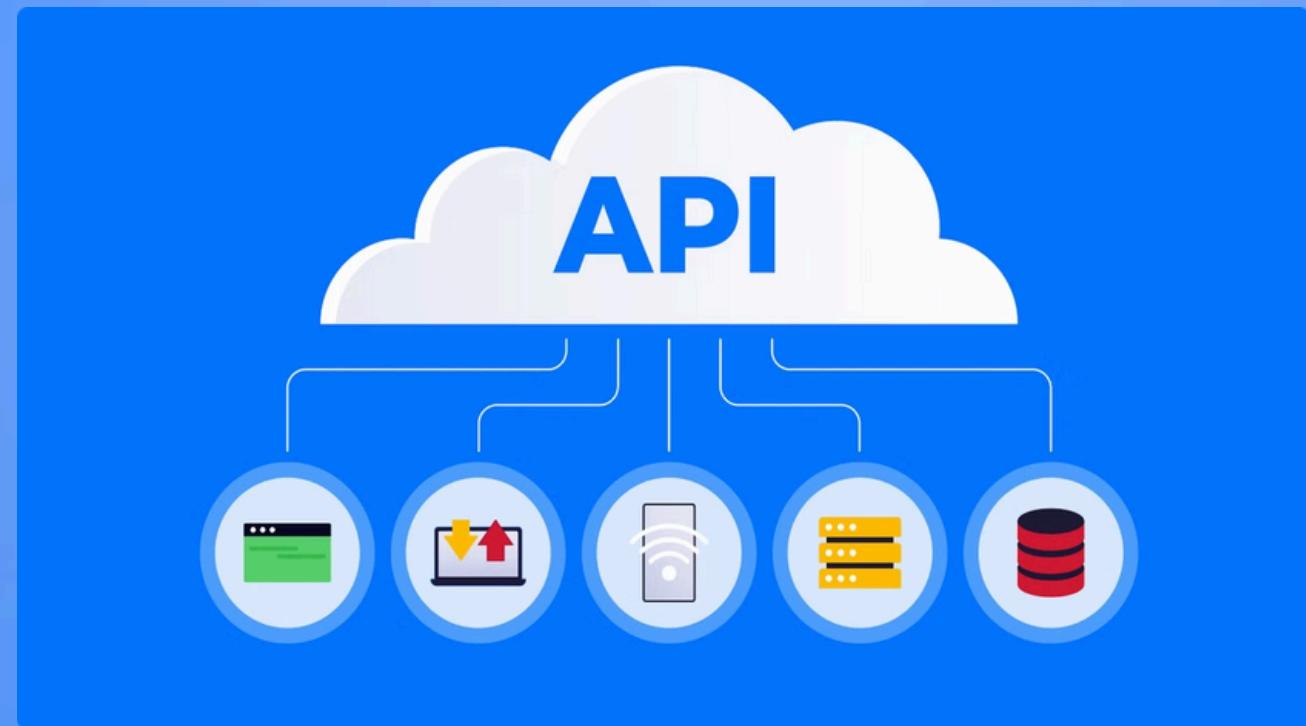
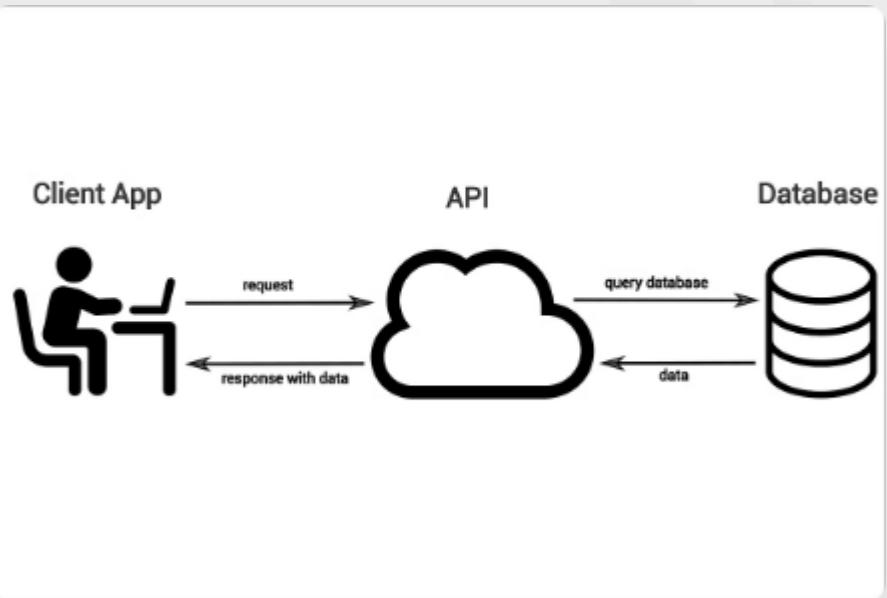


FastAPI

Bem-vindos ao mundo das APIs! Este guia introdutório explorará os fundamentos das APIs, com foco especial no FastAPI para desenvolvedores Python iniciantes. Vamos desvendar os conceitos-chave e práticas essenciais para construir APIs eficientes e seguras.



O que é API?



Introdução às APIs

1

Definição

APIs são interfaces que permitem a comunicação entre diferentes sistemas de software.

2

Importância

Facilitam a integração de serviços e o compartilhamento de dados entre aplicações.

3

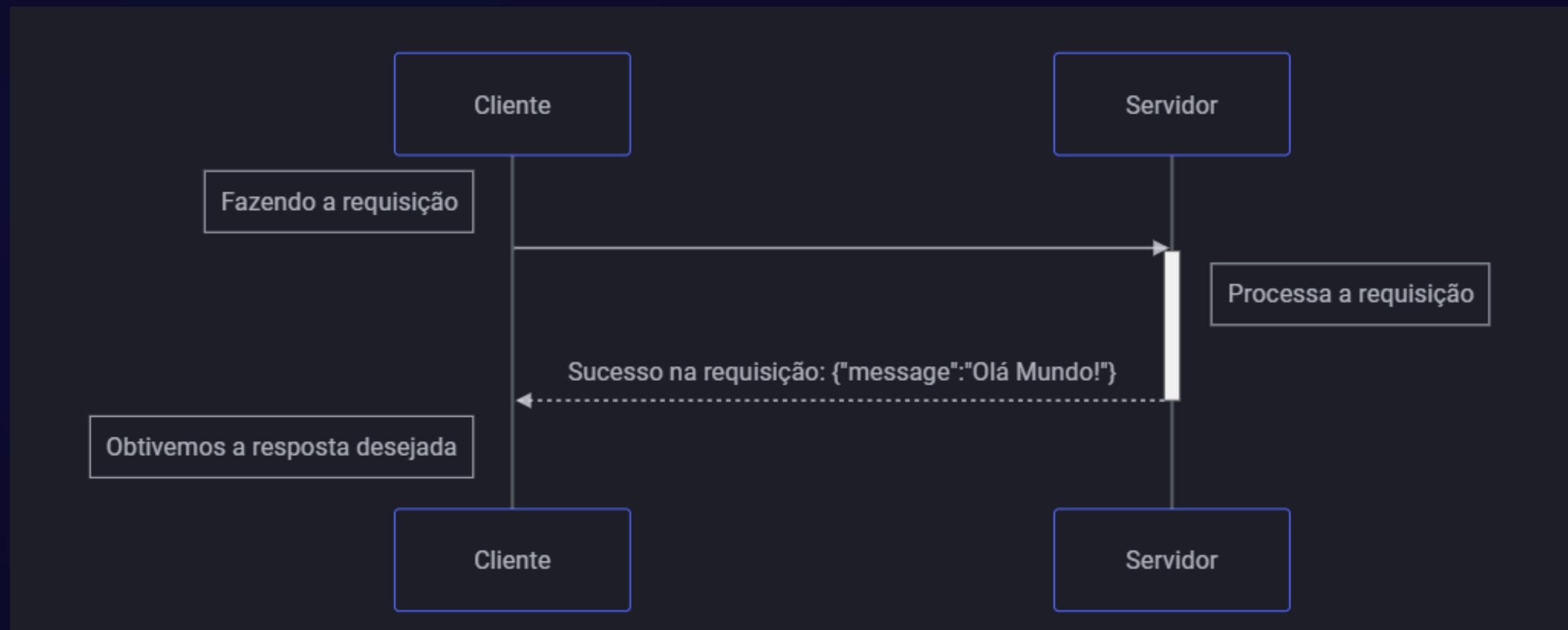
Versatilidade

Podem ser usadas em websites, aplicativos móveis, IoT e muito mais.

Entendendo a WEB - Cliente Servidor

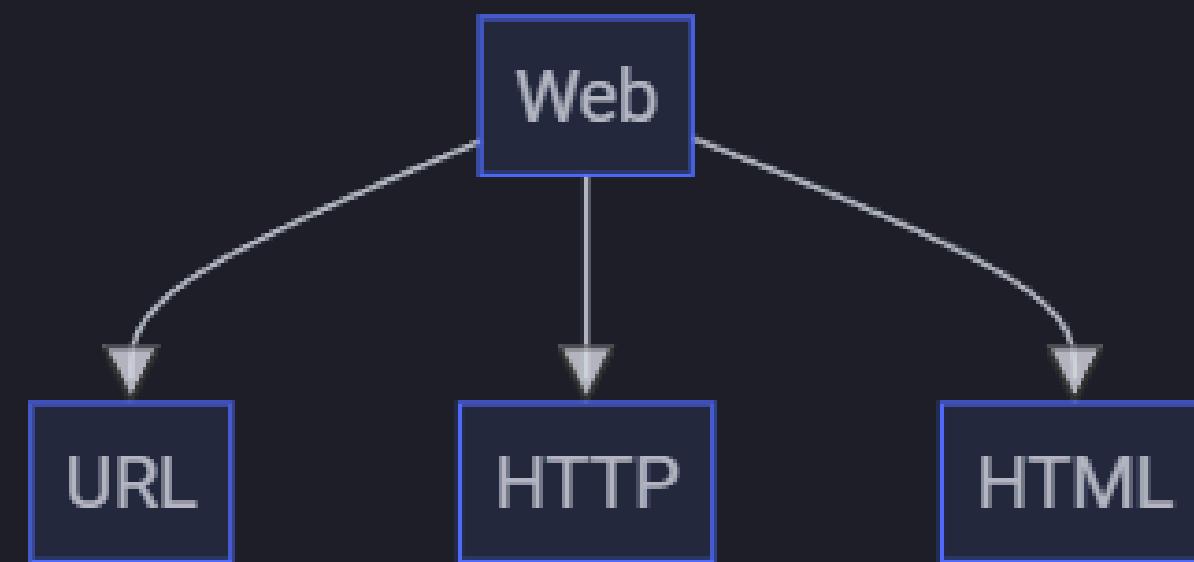


Entendendo a WEB - Cliente Servidor



O modelo padrão da web

Ignorando muita história e diversas camadas de padrões, podemos nos concentrar nos três padrões principais que serão mais importantes para nós agora:



- **URL**: *Localizador Uniforme de Recursos*. Um endereço de rede pelo qual podemos nos comunicar com um computador na rede.
- **HTTP**: um protocolo que especifica como deve ocorrer a comunicação entre dispositivos.
- **HTML**: a linguagem usada para criar e estruturar páginas na web.



Entendendo os Verbos HTTP

GET

Recupera dados

POST

Cria novos recursos

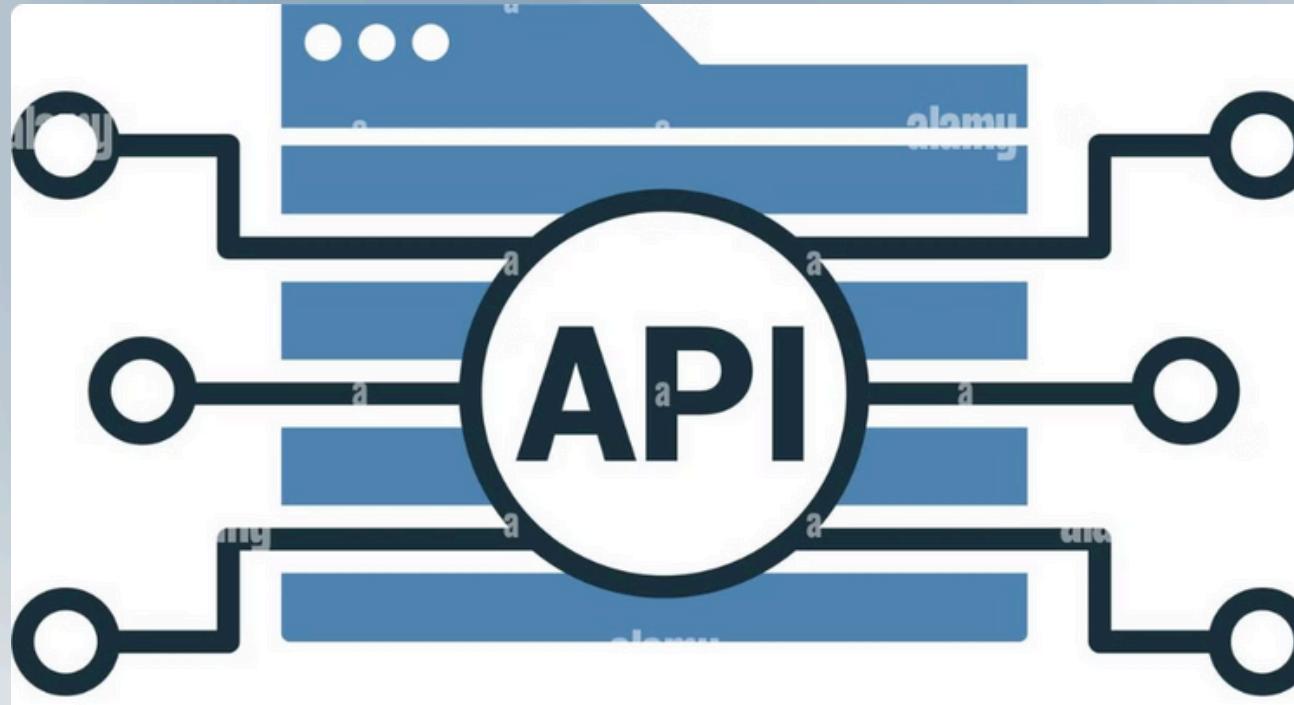
PUT

Atualiza recursos existentes

DELETE

Remove recursos

Exemplos de APIs PÚBLICAS



OpenWeatherMap

Fornece dados meteorológicos atualizados para qualquer localização.



Google Maps

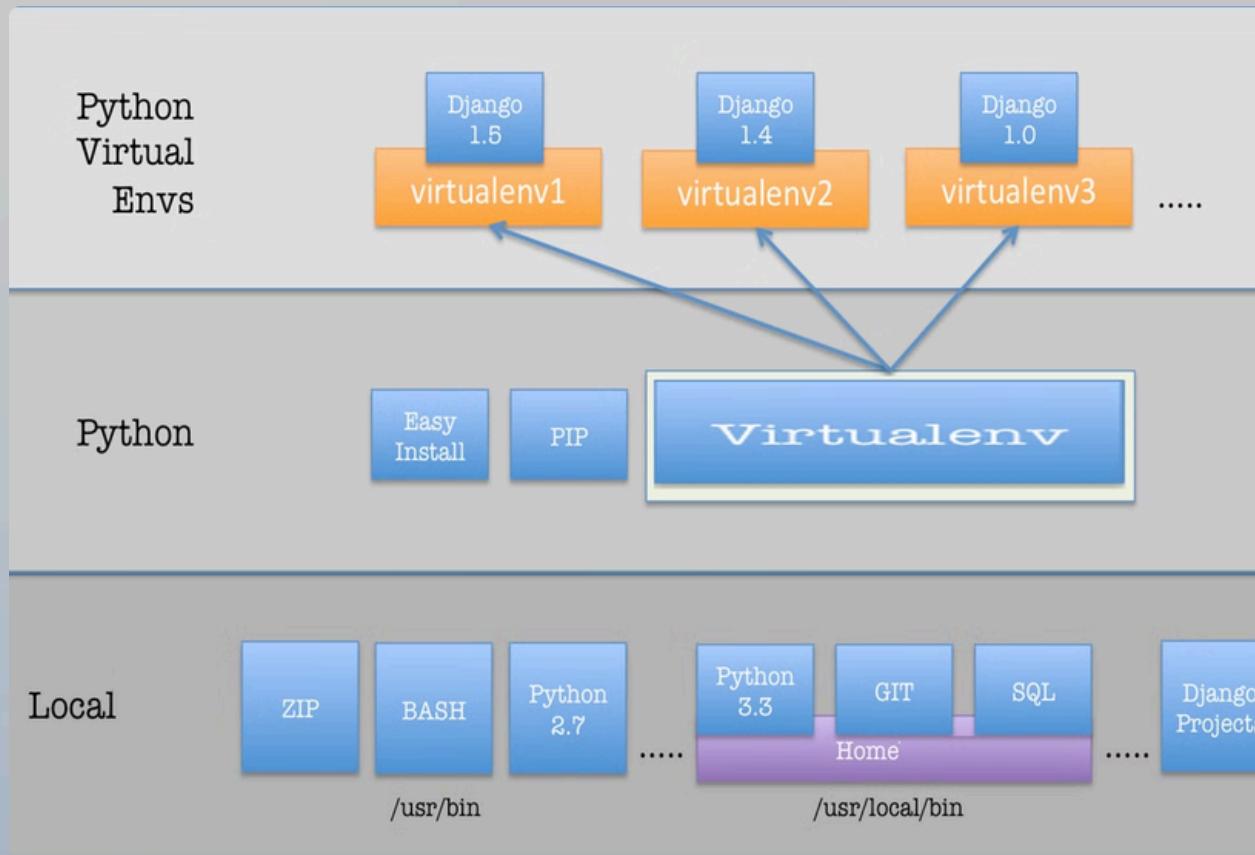
Permite integrar mapas interativos e serviços de localização em aplicativos.



Stripe

Facilita a integração de sistemas de pagamento em lojas online.

Criação e Ativação do Ambiente Virtual



- 1 Criar ambiente**
Execute '`python -m venv nome_do_ambiente`' para criar um novo ambiente virtual.
- 2 Ativar ambiente**
Use '`source nome_do_ambiente/bin/activate`' (Linux/Mac) ou '`nome_do_ambiente\Scripts\activate`' (Windows).
- 3 Verificar ativação**
O nome do ambiente aparecerá no prompt de comando.



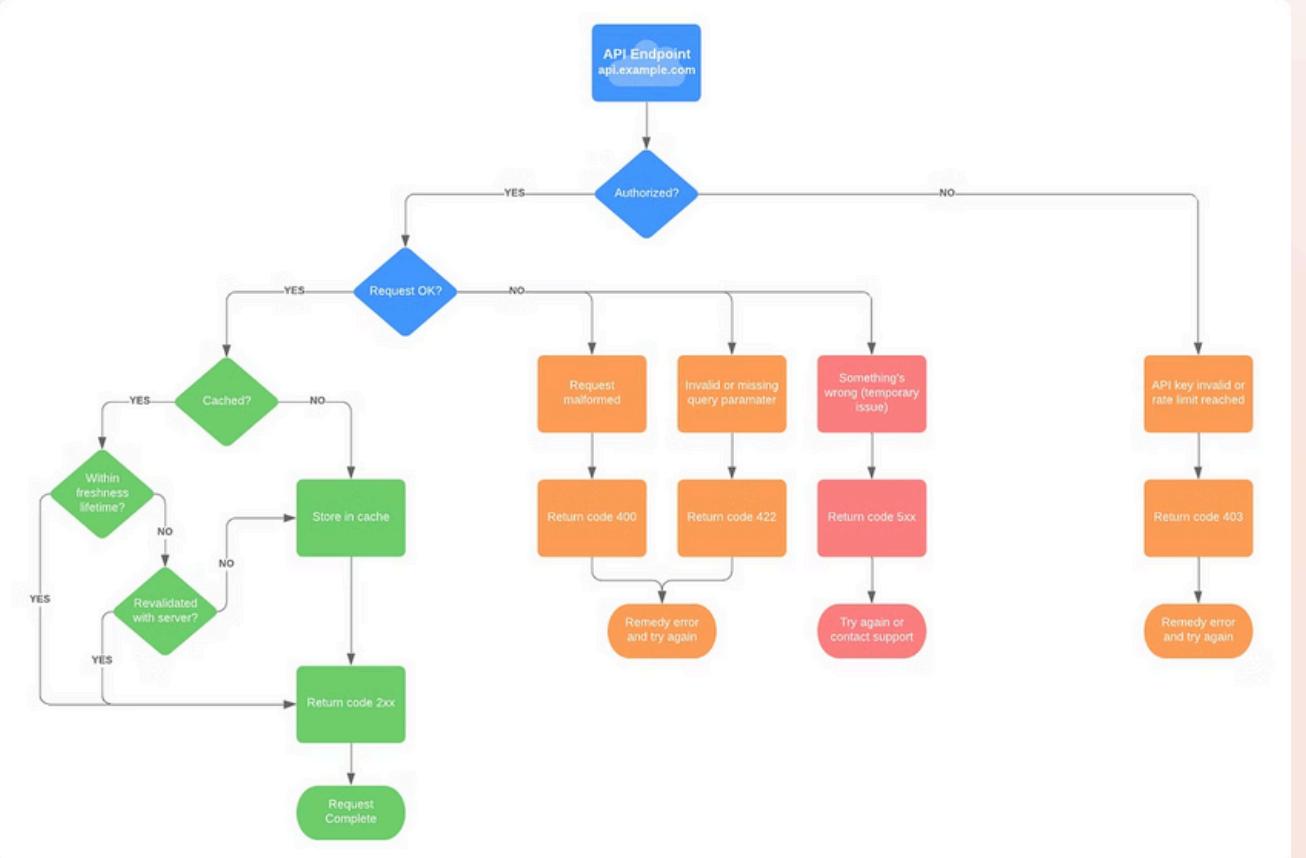
Instalação do FastAPI

Instalar FastAPI

Execute 'pip install fastapi' no ambiente virtual ativado.

Nossa primeira API

Entendendo Endpoints



Definição

Endpoints são URLs específicas onde a API aceita requisições.

1

Estrutura

Geralmente seguem o padrão '/recurso' ou '/recurso/{id}' para operações específicas.

2

Implementação

No FastAPI, use decoradores como `@app.get("/recurso")` para definir endpoints.

3

Entendendo Requests e Responses

Requests

São solicitações enviadas pelo cliente à API. Podem incluir parâmetros, corpo e cabeçalhos.

Responses

São respostas da API ao cliente. Contêm dados solicitados e um código de status.

Status Codes

Indicam o resultado da requisição. Ex: 200 (OK), 404 (Not Found), 500 (Server Error).

Requisição

“Por debaixo dos panos.”

```
GET / HTTP/1.1
Accept: /*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: 127.0.0.1:8000
User-Agent: HTTPie/3.2.2
```

Resposta

```
HTTP/1.1 200 OK
content-length: 24
content-type: application/json
date: Fri, 19 Jan 2024 04:05:50 GMT
server: uvicorn

{
    "message": "Olá mundo"
}
```

Atividade 1

Esta atividade cria uma API simples com três rotas diferentes usando o método GET, que respondem com mensagens de saudação personalizadas. A aplicação FastAPI exibe uma mensagem de boas-vindas geral ou uma saudação específica com base no cargo da pessoa (coordenador ou diretor).

Funcionalidades:

Rota Principal ("/"):

Quando o usuário acessa a URL principal (/), a API retorna uma mensagem com: "Seja bem vindo!".

Rota para Coordenador ("/coordenador"):

Ao acessar a rota /coordenador, a API responde com: "Olá, Coordenador!".

Rota para Diretor ("/diretor"):

Ao acessar a rota /diretor, a API retorna: "Olá, Diretor!".

Status de resposta

No mundo das requisições usando o protocolo HTTP, além da resposta obtida quando nos comunicamos com o servidor, também recebemos um código de resposta (status code).

Os códigos são formas de mostrar ao cliente como o servidor lidou com a sua requisição. Os códigos são divididos em classes e as classes são distribuídas por centenas:

1xx: informativo – utilizada para enviar informações para o cliente de que sua requisição foi recebida e está sendo processada.

2xx: sucesso – Indica que a requisição foi bem-sucedida (por exemplo, 200 OK, 201 Created).

3xx: redirecionamento – Informa que mais ações são necessárias para completar a requisição (por exemplo, 301 Moved Permanently, 302 Found).

4xx: erro no cliente – Significa que houve um erro na requisição feita pelo cliente (por exemplo, 400 Bad Request, 404 Not Found).

5xx: erro no servidor – Indica um erro no servidor ao processar a requisição válida do cliente (por exemplo, 500 Internal Server Error, 503 Service Unavailable).

Status Code

200 OK – Tudo certo! 

Você pediu uma pizza, ela chegou certinha e todo mundo saiu feliz.

201 Created – Algo novo nasceu! 

Você pediu uma pizza única e a pizzaria criou sua invenção com sucesso.

404 Not Found – Não encontramos! 

Você pediu algo estranho e o garçom ficou perdido, pois não existe no cardápio.

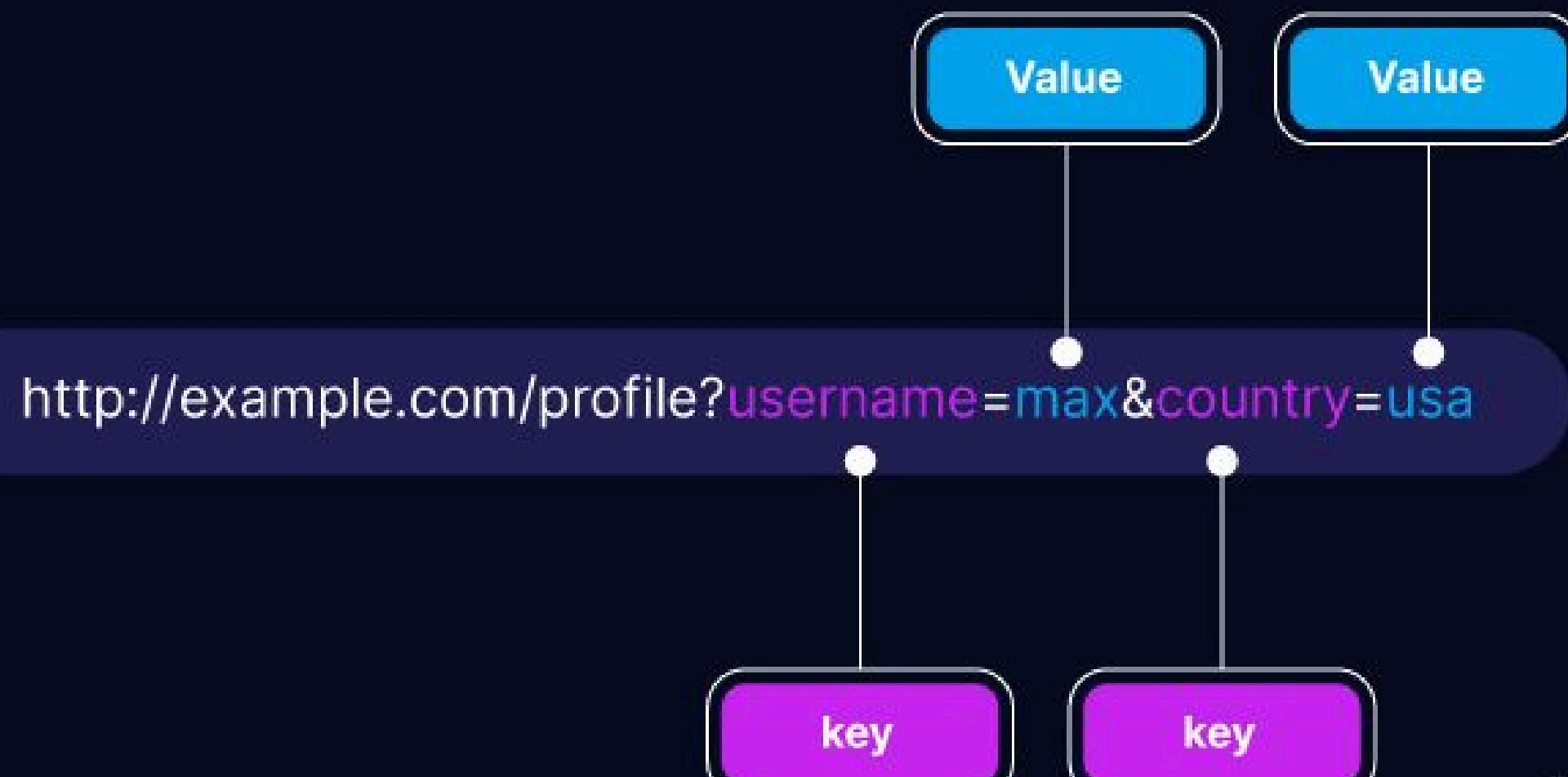
422 Unprocessable Entity – Não dá pra fazer... 

Você pediu uma pizza impossível (tipo de água) e, mesmo entendendo, o garçom não consegue fazer.

500 Internal Server Error – A cozinha pegou fogo! 

Você pediu sua pizza, mas algo deu errado na cozinha e nada pode ser servido agora.

Query Parameters



Atividade 2

Criar um endpoint que retorna uma mensagem para número par, outra para número ímpar.

Funcionalidades:

Rota Principal ("/par-ou-ímpar"):

Quando o usuário acessa a URL principal (/par-ou-ímpar), a API retorna uma mensagem.

Atividade 3

Converter temperaturas de Celsius para Fahrenheit

- Rota que recebe uma temperatura em Celsius e retorna o valor convertido para Fahrenheit.