AULA 01

FASTAPI

API E INTRODUÇÃO AO FASTAPI

O QUE VEREMOS HOJE

01 API

02 HTTP

03 MÉTODOS HTTP

04 REQUEST E RESPONSE

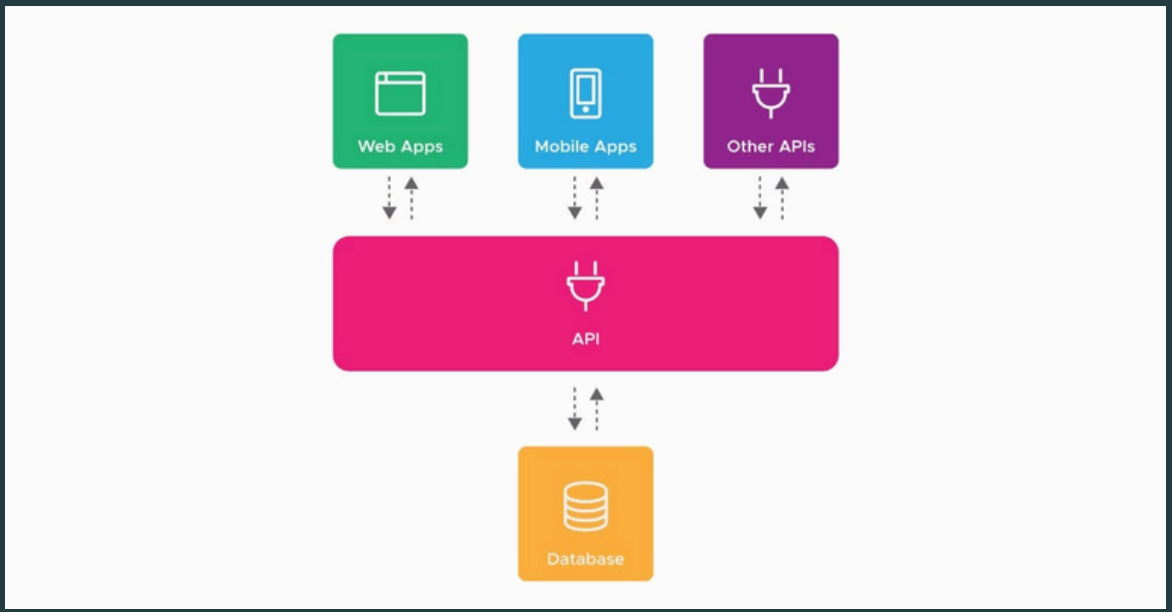
05 JSON

06 INTRODUÇÃO AO FASTAPI

API - APPLICATION PROGRAMMING INTERFACE

É um intermediário de software que permite que sistemas se comuniquem entre si. Elas definem regras e formatos específicos para que as informações sejam trocadas de forma organizada e eficiente.

Através de APIs, um sistema pode acessar recursos de outro sistema, como dados, funcionalidades ou serviços.



TIPOS DE API

APIs RESTful (Representational State Transfer)

São as mais comuns, utilizando o protocolo HTTP para comunicação. Elas são conhecidas por sua simplicidade e facilidade de uso, sendo amplamente adotadas em diversos cenários.

APIs GraphQL

Criada pelo Facebook, é uma das mais modernas e permite ao cliente solicitar dados específicos, reduzindo a sobrecarga de informações.

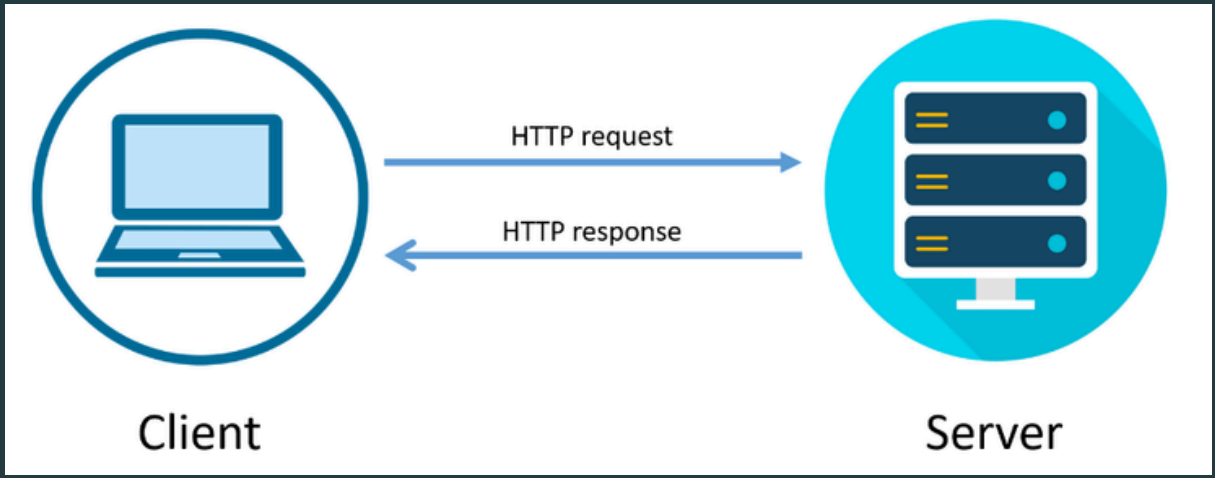
APIs SOAP (Simple Object Access Protocol)

Baseada em XML, esse tipo de API define um padrão para comunicação de mensagens entre sistemas, oferecendo mecanismos para validação de dados, controle de acesso e criptografia, entre outras ferramentas com alto nível de segurança.

O QUE É HTTP?

HTTP (Hypertext Transfer Protocol) é o protocolo de **comunicação** que permite a troca de informações entre navegadores web e servidores web. É uma base fundamental para a comunicação entre APIs e aplicativos.

O HTTP define as regras de comunicação, como a estrutura das requisições e respostas, os códigos de status e os tipos de conteúdo. APIs usam o HTTP para transmitir dados e comandos entre servidores e aplicativos.



MÉTODOS HTTP

GET

É utilizado para buscar informações de um servidor. É o método mais básico e comum, e retorna dados em formato de texto, normalmente em HTML, JSON ou XML.

POST

O método POST é utilizado para enviar dados para um servidor, por exemplo, para criar um novo recurso, como um novo usuário em um site.

PUT

O método PUT é utilizado para atualizar um recurso existente em um servidor, por exemplo, para editar as informações de um usuário.

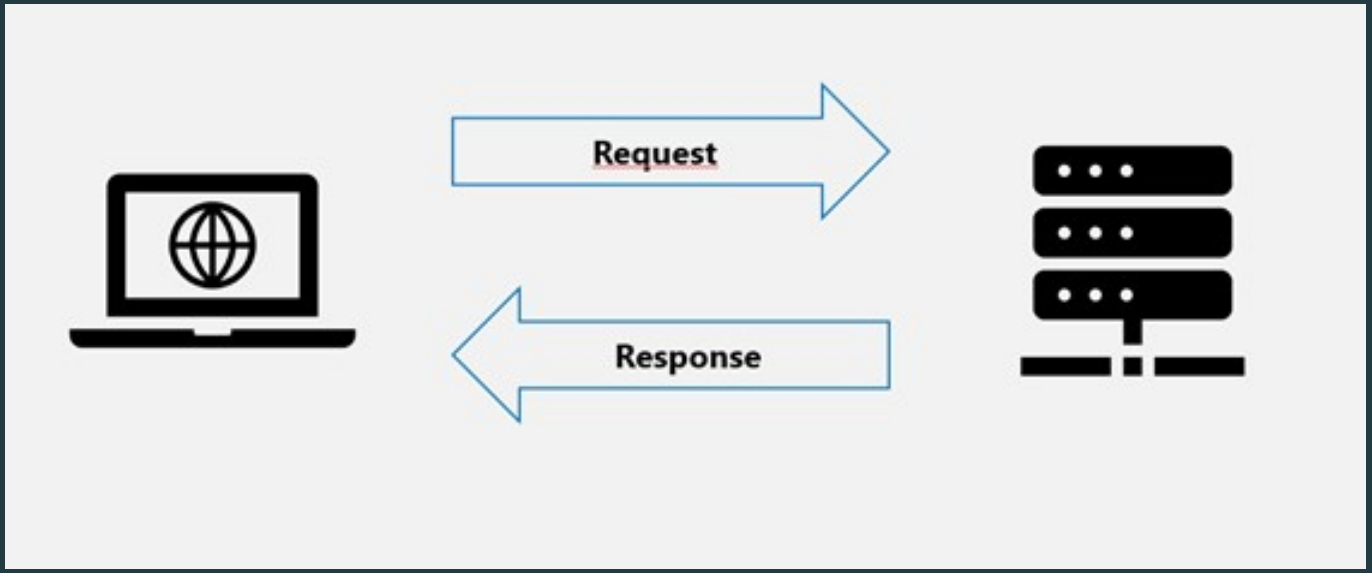
DELETE

O método DELETE é utilizado para remover um recurso de um servidor, por exemplo, para excluir um usuário de um sistema.

REQUEST

Uma request é uma solicitação enviada de um sistema para outro, geralmente através da internet. Essa solicitação contém informações sobre o que o sistema requisitante deseja obter do outro sistema.

A request geralmente inclui o método HTTP, a URL do recurso solicitado, cabeçalhos e um corpo com os dados da solicitação.



REQUEST - COMPONENTES

Método HTTP

O método indica a ação desejada, como obter dados (GET) ou enviar informações (POST).

URL

A URL identifica o recurso específico que está acessando. Ela é formada pelo endereço do servidor, o nome da API e um identificador único do recurso.

Headers

Os headers fornecem informações adicionais sobre a request, como o tipo de conteúdo que está sendo enviado.

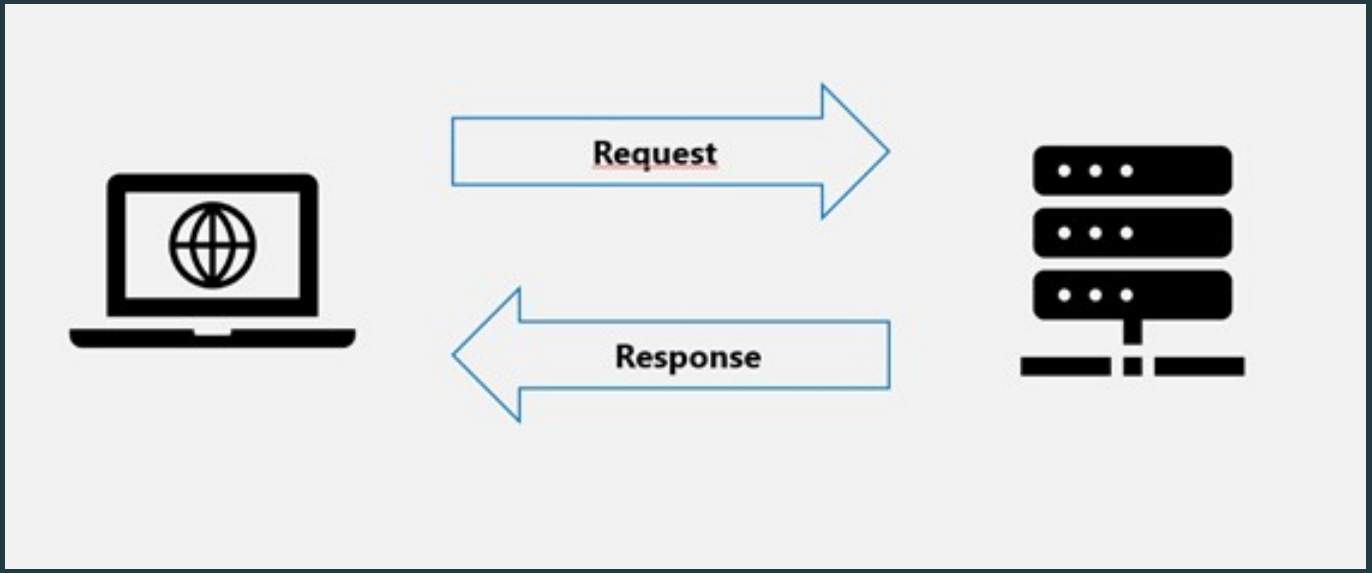
Body

O body contém os dados que está sendo enviado para o servidor. O conteúdo do body varia de acordo com o método HTTP.

RESPONSE

Response é a resposta do servidor à request. No response irá conter as informações sobre o sucesso ou falha da requisição, dados solicitados e outros detalhes.

Assim como no request, o response também possui componentes em sua estrutura.



RESPONSE - COMPONENTES

Status Code

O status code é um número que indica o sucesso ou o erro da requisição. 200 significa sucesso, 404 significa que a página não foi encontrada, e 500 indica um erro do servidor.

Headers

Os headers são metadados que fornecem informações adicionais sobre a resposta, como o tipo de conteúdo.

Body

O body da resposta contém o conteúdo real da resposta, como dados, texto ou imagens. O formato do body depende da API e pode ser JSON, XML ou outros.

JSON

JSON (JavaScript Object Notation) é um formato leve de troca de dados e usado para comunicação entre sistemas, especialmente em aplicações web.

A estrutura do JSON é similar a pares chave-valor, mas ele é sempre uma string formatada, podendo ser convertida para tipos nativos, como dicionários, em várias linguagens de programação.

OBS: No JSON só é possível usar aspas duplas.

```json
{
    "nome": "José Maria",
    "email": "jose@gmail.com",
    "idade": 30,
    "cidade": "São Paulo",
    "estado": "SP",
    "status": "Ativo"
}
```

ATIVIDADE PRÁTICA

Utilizando a API Dummy JSON e faça as seguintes requisições:

- uma requisição para buscar a lista de produtos disponíveis.
- uma requisição buscar um único produto.
- uma requisição para adicionar um novo produto à API.

Verifique a documentação para fazer as requisições.

FASTAPI

FastAPI é um framework Python moderno e de alto desempenho, ideal para construir APIs web.

Ele oferece recursos como tipagem estática, validação de dados, documentação automática e integração com bancos de dados.

O FastAPI foi projetado para ser rápido, fácil de usar e eficiente.

AMBIENTE VIRTUAL

Um ambiente virtual é uma ferramenta que permite a criação de ambientes isolados para cada projeto. Cada projeto pode ter suas próprias dependências, evitando conflitos e garantindo que as bibliotecas necessárias estejam disponíveis.

No Python isso é feito por meio da venv que é um um diretório separado que contém uma cópia independente do interpretador Python e um conjunto de pacotes

COMO INICIAR UM AMBIENTE VIRTUAL

Para criar um ambiente virtual (venv) é só digitar o comando no terminal:

python -m venv venv

Esse comando deve criar uma pasta chamada venv no diretório.

Para ativar o ambiente virtual temos que usar o seguinte comando:

- Windows: **venv/Scripts/activate**
- macOS/Linux: **source venv/bin/activate**

FASTAPI - CONFIGURAÇÃO

Com a venv ativada, é possível instalar o FastAPI e o Uvicorn.

Para instalar, rode o comando: **pip install fastapi uvicorn**

```
(venv) aline@aline-Aspire-A515-54:~/WorkArea/Infinity/aula_01$ pip install fastapi uvicorn
Collecting fastapi
  Downloading fastapi-0.114.2-py3-none-any.whl (94 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 94.0/94.0 kB 946.1 kB/s eta 0:00:00
Collecting uvicorn
  Downloading uvicorn-0.30.6-py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.8/62.8 kB 6.2 MB/s eta 0:00:00
```

Para criar o arquivo requirements.txt usamos o comando: **pip freeze > requirements.txt**

Este arquivo lista todas as dependências do projeto.

```
(venv) aline@aline-Aspire-A515-54:~/WorkArea/Infinity/aula_01$ pip freeze > requirements.txt
(venv) aline@aline-Aspire-A515-54:~/WorkArea/Infinity/aula_01$
```

ATIVIDADE PRÁTICA

Adicione um novo endpoint GET no arquivo main.py que retorne uma lista de dicionários com informações sobre o produto seguindo esse exemplo:

- id: 1
- nome: Produto 1
- preço: 28,90
- estoque: 50

Teste no Postman e verifique se retornou a lista.

ATIVIDADE PRÁTICA

Adicione um novo endpoint GET no arquivo main.py que retorne o total de livros na lista e uma lista de dicionários com informações livros seguindo esse exemplo:

- id: 1
- título: Livro 1
- preço: 45,90
- autor: José da Silva

Teste no Postman e verifique se retornou a lista e o total de livros da lista.