

ESTRUTURA DE API



HTTP/REST



HTTP: (Protocolo de Transferência de Hipertexto) é como um intermediário entre você e a internet. Ele permite que seu computador peça informações (como um site ou um vídeo) e a internet forneça essas informações de volta.

- Assim como você pede um livro ao bibliotecário, o HTTP pede informações ao "bibliotecário da internet", que é o servidor.



HTTP/REST



REST: Transferência de Estado Representacional, é um estilo de arquitetura usado para facilitar a comunicação entre sistemas na web. Imagina um garçom que leva seus pedidos (solicitações) para a cozinha (servidor) e traz de volta seu prato (dados).

- Ele usa URLs para acessar recursos e métodos específicos para realizar operações, tornando fácil para os desenvolvedores criar e manter aplicações web.

CONCEITOS HTTP



MÉTODOS HTTP

GET: Recupera informações do servidor.

POST: Envia novas informações para o servidor.

PUT: Atualiza informações existentes no servidor.

DELETE: Remove informações do servidor.

PATCH: Atualiza parcialmente as informações no servidor.

CÓDIGOS DE STATUS HTTP

200 OK: A solicitação foi bem-sucedida.

201 Created: A solicitação foi bem-sucedida e um novo recurso foi criado como resultado.

204 No Content: A solicitação foi bem-sucedida, o corpo da resposta está vazio.

400 Bad Request: A solicitação não pôde ser entendida ou estava faltando parâmetros necessários.

401 Unauthorized: A autenticação falhou ou ainda não foi fornecida.

403 Forbidden: A parte autenticada não tem acesso ao recurso solicitado.

404 Not Found: O recurso solicitado não pôde ser encontrado.

503 Service Unavailable: O serviço está temporariamente indisponível.

PET

DEMPE

DELET

DELETE

COMPONENTES DE UMA API

Recursos: Representam objetos de domínio específicos na API

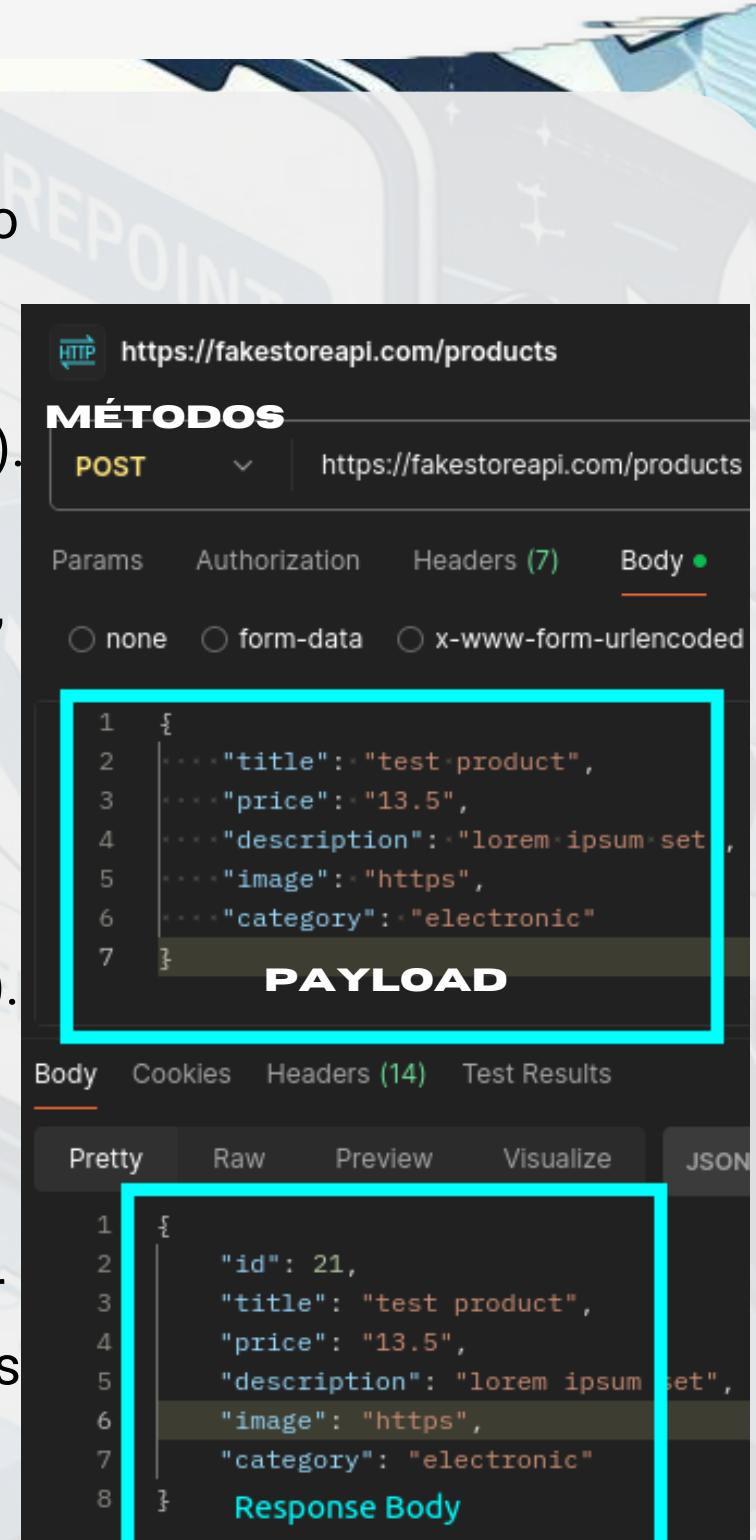
Métodos: Ações que podem ser realizadas nos recursos(ex.: GET, POST, PUT, DELETE).

Parâmetros: Usados para personalizar solicitações, podendo ser incluídos na URL, no cabeçalho ou no corpo da solicitação (Ex: Path, Query , Payload)

Cabeçalhos: Usados para transmitir informações adicionais sobre a solicitação ou a resposta (ex.: conteúdo, autenticação).

Corpo: Usado para enviar dados do cliente para o servidor, geralmente em formatos como JSON.

Respostas: Dados retornados pelo servidor para o cliente, contendo tanto os resultados da solicitação quanto informações adicionais como status da solicitação e cabeçalhos de resposta.



COMPONENTES DE UMA API

ENDPOINT: URL onde a API pode ser acessada.

EXEMPLO:

Get all products URL: <https://fakestoreapi.com/products>

The screenshot shows a POSTMAN interface with the following details:

- URL:** https://fakestoreapi.com/products
- Method:** GET
- Headers:**
 - Host: fakestoreapi.com
 - User-Agent: PostmanRuntime/7.37.3
 - Accept: */*
 - Accept-Encoding: gzip, deflate, br
 - Connection: keep-alive
- Status:** 200 OK
- Body (JSON):**

```
1 [ { 2   "id": 1, 3   "title": "Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops", 4   "price": 109.95, 5   "description": "Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday", 6   "category": "men's clothing", 7   "image": "https://fakestoreapi.com/img/81fPKd-2AYL._AC_SL1500_.jpg", 8   "rating": { 9     "rate": 3.9, 10    "count": 120 11  } 12 }, 13 ]
```

CABEÇALHOS HTTP



CABEÇALHOS DE SOLICITAÇÃO

- **User-Agent:** Informa sobre o software do cliente (ex: navegador, versão, sistema operacional)
- **Accept:** Informa os tipos de mídia que o cliente pode processar
- **Authorization:** Usado para autenticação e autorização (credenciais, tokens de autenticação, chaves de API).

CABEÇALHOS DE RESPOSTA

- **Server:** Informa sobre o software usado pelo servidor
- **Content-Type:** Informa o tipo de mídia do corpo da resposta
- **Set-Cookie:** Usado pelo servidor para enviar cookies ao cliente

CABEÇALHOS HTTP



CABEÇALHOS GERAIS

- **Cache-Control:** Especifica diretivas de cache para cliente e servidor
- **Date:** Contém a data e a hora em que a mensagem foi enviada

CABEÇALHOS DE ENTIDADE

- **Content-Length:** Informa o tamanho, em bytes, do corpo da solicitação/resposta
- **Content-Encoding:** Especifica o método de codificação usado no corpo da solicitação/resposta

Credentials:	
Access-Control-Allow-Origin:	https://www.amazon.com.br
Access-Control-Expose-Headers:	x-amzn-RequestId,x-amzn-ErrorBoundary
Headers:	
Connection:	keep-alive
Content-Encoding:	gzip
Content-Length:	22
Content-Type:	application/json
Date:	Tue, 09 Apr 2024 17:03:10 GMT
Server:	Server
Strict-Transport-Security:	max-age=47474747; includeSubDomains
Vary:	Origin,Content-Type,Accept-Encoding
X-Amz-Rid:	0DVFRYF238DC3F66KAAJ
X-Amzn-Requestid:	4a2c1a25-e0e7-417d-a46a-385c
Request Headers	<input type="checkbox"/> Raw
Accept:	/*
Accept-Encoding:	gzip, deflate, br, zstd
Accept-Language:	pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7
Connection:	keep-alive
Content-Length:	160
Content-Type:	application/json
Host:	unagi.amazon.com.br
Origin:	https://www.amazon.com.br
Referer:	https://www.amazon.com.br/
Sec-Ch-Ua:	"Not A(Brand";v="99", "Google Chrome";v="118.0.5931.132", "Chromium";v="121"
Sec-Ch-Ua-Mobile:	?0
Sec-Ch-Ua-Platform:	"Linux"
Sec-Fetch-Dest:	empty
Sec-Fetch-Mode:	cors

ABA O DEVTOOLS(F12), CLICK EM NETWORK,
CLIQUE EM UMA CHAMADA E ANALISE A TAB
(HEADERS)

API

TIPOS DE DADOS DE API

XML

XML (eXtensible Markup Language): Estrutura hierárquica e baseada em tags, utilizada para armazenar e transportar dados.

- XML: Baseado em Documento, Suporta Atributos, Menos eficiente na análise de dados

JSON

JSON (JavaScript Object Notation): Formato de dados leve e fácil de ler e escrever, utilizado para estruturar dados simples

- JSON: Baseado em Dados, Não suporta Atributos, Mais eficiente na análise de dados

JSON

```
{ "user": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "johndoe@example.com",  
    "status": "active"  
}
```

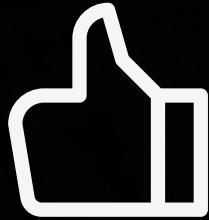
XML

```
<user>  
  <id>1</id>  
  <name>John Doe</name>  
  <email>johndoe@example.com</email>  
  <status>active</status>  
</user>
```



📢 *Quer aprender mais?
Acompanhe nossa série sobre testes de API!*

Interaja com a postagem



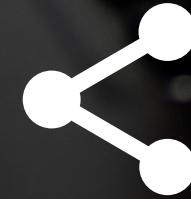
Curte



Comente



Envie



Compartilhe