

Introdução ao mundo das APIs



Diversas ações que executamos em sites na web precisam buscar informações em servidores para nos dar uma resposta, para essa interação entre navegador e servidor acontecer, é utilizado um protocolo chamado HTTP (HyperText Transfer Protocol).

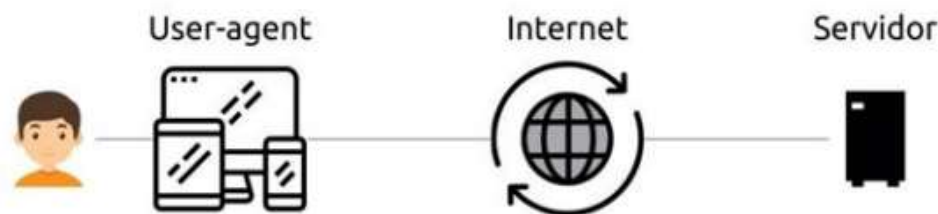
Os servidores esperam por solicitações (**requisições HTTP**), quando elas chegam, são processadas e após isso dão um retorno ao navegador da web, por meio de uma **resposta HTTP**. Neste cenário temos 2 partes se comunicando: **cliente e servidor**.



Cliente



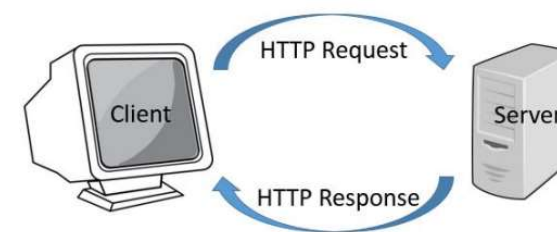
- É o cliente que geralmente inicia a comunicação com o servidor.
- O cliente mais conhecido é o navegador.
- Após enviar uma requisição, o cliente aguarda até que haja uma resposta do servidor e então a comunicação se encerra.
- Para obter um novo recurso do servidor o cliente deve iniciar uma nova requisição.



Servidor



- O papel do servidor web é receber uma requisição e devolver uma resposta para o cliente.
- O servidor é composto por um ou mais computadores com um único endereço como `http://www.google.com.br`.
- O servidor hospeda documentos e programas que atendem as requisições dos clientes e as processam. Chamamos esses programas de web server, ou servidor web.

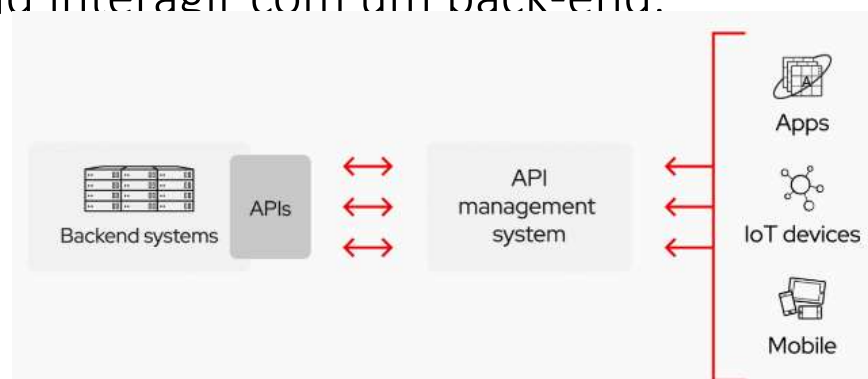


API - Interface de Programação de Aplicações



É um conjunto de rotinas que define como o cliente e o servidor podem interagir. Nela são definidos as URLs, métodos, parâmetros, formatos de dados (JSON, XML, etc) e outros detalhes que os desenvolvedores precisam seguir para interagir corretamente com o servidor.

Por meio do uso de APIs, é possível que diferentes aplicações conversem entre si, por exemplo, um front-end interagir com um back-end.



Protocolo HTTP



Os clientes se comunicam com servidores utilizando o protocolo HTTP. Para isso são enviadas requisições para uma API, as quais são compostas por elementos como:

- URL: indica o alvo da requisição (recurso);
- Método: define a ação que deve ser executada, como obter, excluir ou criar um recurso;
- Informações adicionais: para fornecer algum filtro ou parâmetro no recurso, por exemplo ordenar os resultados de forma decrescente.

Identificadores de Recursos



URI (Identificador uniforme de recurso): é uma cadeia de caracteres usada para identificar um recurso na internet.

URL (Localizador uniforme de recursos): é um **tipo de URI** que define um endereço do recurso na rede. Os componentes típicos de uma URL incluem o esquema (http, https), o host (www.dominio.com), o caminho (/usuarios), entre outros.

Exemplos de recursos:

URI: `http://www.dominio.com/usuarios`

URI: `http://www.dominio.com/produtos/1`

Métodos HTTP



Os métodos vão definir o tipo de ação a ser executada, por exemplo:

GET - para recuperar dados

POST - para enviar dados

PUT - para atualizar dados

DELETE - para excluir dados



Embora esses métodos possam ser descritos como substantivos, eles também são comumente referenciados como HTTP Verbs (Verbos HTTP)

ELEMENTOS DE RETORNO HTTP



Quanto temos um retorno de requisição, recebemos um status de retorno HTTP, que demonstra o resultado da requisição. Temos diversos tipos de retorno que podem ocorrer quando chamamos uma API.

Respostas de informação (100-199),
Respostas de sucesso (200-299),
Redirecionamentos (300-399)
Erros do cliente (400-499)
Erros do servidor (500-599).

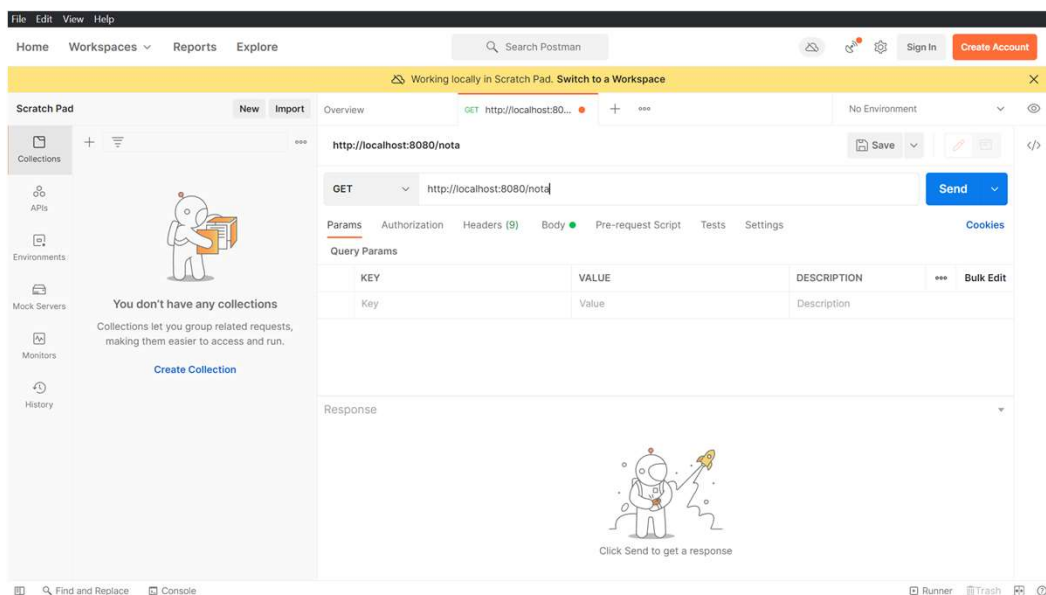


CÓDIGOS DE RETORNO HTTP



- 200 -> OK: o servidor retorna os dados solicitados com sucesso
- 201 -> Created: um novo recurso foi criado com sucesso
- 204 -> No content: solicitação foi correta, mas o servidor não retorna conteúdos
- 400 -> Bad Request: solicitação contém dados inválidos ou está mal formada
- 401 -> Unauthorized: cliente deve fornecer autenticação para acessar o recurso
- 403 -> Forbidden: o acesso ao recurso é proibido.
- 404 -> Not found: recurso solicitado não foi encontrado no servidor.
- 405 -> Method not allowed: método incorreto, trocou GET por POST, por exemplo
- 500 -> Internal Server error: servidor encontrou um erro interno ao processar a solicitação
- 503 -> Service Unavailable: servidor não está disponível para lidar com a solicitação devido a sobrecarga ou manutenção

POSTMAN



Postman é um software que nos permite enviar requisições para APIs e visualizar as respostas.

É muito utilizado para testar APIs sem a necessidade de utilizar um front-end.

Para utilizar o Postman, criamos primeiramente uma coleção (collection) para salvar e organizar nossas requisições.

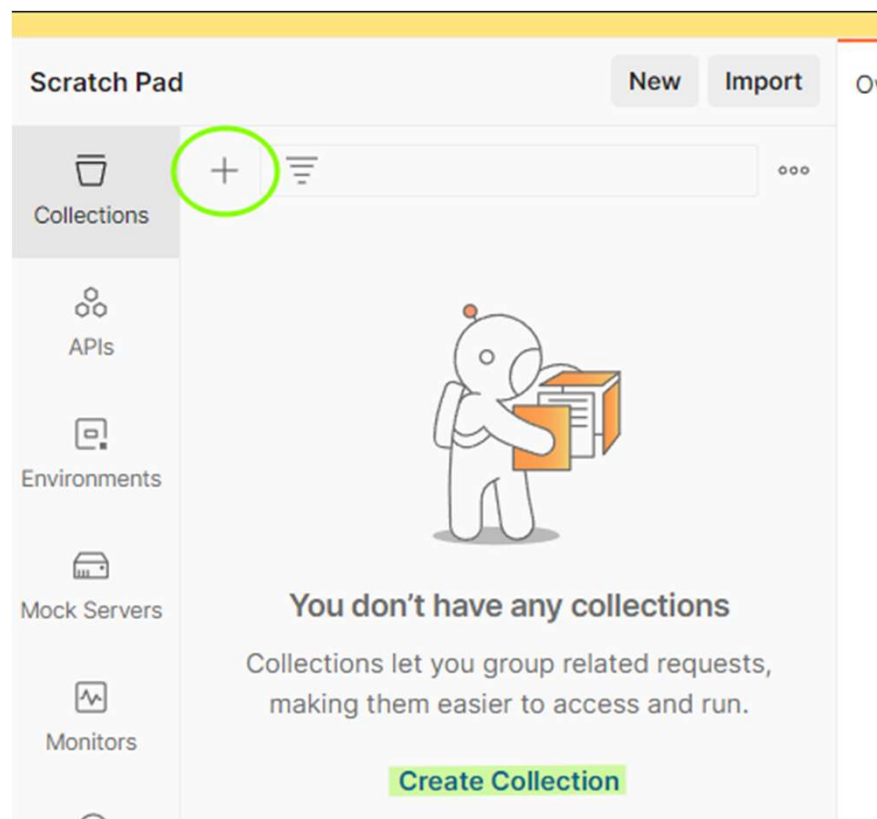
POSTMAN



Para conhecer a ferramenta, vamos interagir com uma API de estudo, que pode ser acessada pela URL:

<https://jsonplaceholder.typicode.com>

Crie a sua collection dentro do Postman para começar a criar requisições:



POSTMAN



Dentro da collection podemos começar a criar os requests que queremos enviar para a API que estamos testando.



POSTMAN



Então podemos escolher se é um request do tipo GET, POST, PUT, etc. E definir o endereço que iremos acessar (normalmente chamado de [endpoint](#)).



GET



GET /posts

Um request com método GET nos permite consultar dados e podemos passar parâmetros de query ou variáveis no caminho da URL.

No método GET, os valores vão todos na URL em formato de texto, por isso não é comum passar informações sensíveis por esse método.

O método POST é mais seguro, pois possui uma parte extra chamada body que pode ser criptografado.

GET



<https://jsonplaceholder.typicode.com/posts/100>



Personalizando as consultas com parâmetros

Path Parameters são utilizados para **identificar** um recurso.

Query Parameters são utilizados para **filtrar** esses recursos.

Exemplos:

GET com path `/produtos/{id}` trará o produto com determinado id

GET com query `/produtos?precoMin=50&precoMax=150` filtrará os produtos com preço entre 50 e 150



Path Parameters

GET /posts/**1**

Path parameters fazem parte da própria estrutura do caminho da URL e são usados para identificar recursos específicos na API, a partir do uso de / + informação.



Query Parameters

GET /posts?userId=1

Os Query Parameters são parâmetros que não fazem parte do caminho da URL, mas sim da chamada que está sendo realizada. Normalmente são utilizados para **realizar filtros, ordenar ou paginar resultados** na chamada.

São mais comuns em chamadas GET, mas podem estar em todas as outras chamadas também.

POST



POST /posts

O POST é um método utilizado para enviar/cadastrar informações.

Normalmente essas informações são passadas como um JSON no corpo da requisição (body).

O body geralmente é um JSON, mas pode possuir o formato XML ou outros.



PUT



PUT /posts/1

O método PUT é utilizado para atualizar todo um objeto. O novo objeto vai no body da requisição.

Não necessariamente todos os campos são alterados, mas enviamos o objeto completo na requisição.



DELETE



DELETE posts/1

O DELETE é utilizado para apagar um registro de uma API, sendo assim o DELETE deve ser utilizado com cautela pois pode apagar dados necessários para o funcionamento do sistema.

Esse método sempre deve ter verificações e lógicas para prevenir o DELETAR de informações de forma incorreta.

DELETE	▼	https://jsonplaceholder.typicode.com/posts/1
--------	---	---