



NO. DATE
**A
P
I**

pplication
rogramming
nterface

Trocar informações
entre sistemas

Dados que não precisam
de acesso completo

APIs

Interface de programação de aplicações

As APIs são um conjunto de padrões que fazem parte de uma interface. As APIs permitem a criação de plataformas de maneira mais simples e prática para desenvolvedores. A partir de APIs, é possível criar softwares, aplicativos, programas e plataformas diversas.

front-end

back-end

mesa cliente

cozinha

garçon

API

Formato de arquivo leve

pode ser usado por
várias linguagens

JSON

exemplo.json

```
{  
  "nome" : "Manoel Gomes",  
  "musica" : "Caneta Azul"  
}
```

Joy Assis

***Aceita todo
tipo de dado**

API HTTP

HTTP (protocolo de transferência de hipertexto) é o protocolo . Este é o protocolo de aplicativo usado para transferir os dados pela Web. O HTTP usa a arquitetura cliente-servidor. O cliente envia uma solicitação http ao servidor. O servidor processa a solicitação e envia de volta a resposta http. Todo o pedido mundial usa o protocolo http. Quando você carrega o site no navegador, está fazendo uma solicitação http ao servidor.

API REST

REST significa transferência de estado representacional. REST não é um padrão ou uma especificação. REST é um estilo arquitetônico . REST não é um protocolo de comunicação como o HTTP. HTTP é um contrato, um protocolo de comunicação e REST é um conceito .

REST aproveita o protocolo HTTP para transferir os dados pela Internet. REST implica uma série de restrições sobre como o Servidor e o Cliente devem interagir.

Consultando API com axios

axios é que ele é um cliente HTTP baseado em promessas totalmente agnóstico, ou seja, que não depende de frameworks e bibliotecas. Ele é super maleável, podendo ser utilizado do lado do cliente (React, Vue, Svelte, etc) e do lado do backend (express, nest, etc).

Passo 1 - instalar a biblioteca axios pelo terminal

```
$ npm install axios
```



**leia a documentação
do axios**

<https://axios-http.com/ptbr/docs/intro>



conceitos que vamos usar para fazer o consumo

Assync/Awaiy

Os métodos Async e Await são recursos do JavaScript que são comumente usados em conjunto para lidar com código assíncrono de forma mais legível e fácil de manter.

O método **async** é usado para declarar uma função assíncrona que retorna uma Promessa. A Promessa representa um valor que pode não estar disponível no momento em que a função é chamada, mas que será fornecido em algum momento no futuro.

O método **await** é usado para pausar a execução de uma função assíncrona até que uma Promessa seja resolvida. Quando uma Promessa é resolvida, o valor retornado pela Promessa é atribuído à variável na qual o await foi usado.

Aqui vamos usar esses métodos para lidar com chamadas da API. Quando se trata de chamadas de API, por exemplo, Async e Await podem ser usados para aguardar a resposta da API antes de prosseguir com o restante do código.

ComponentDidMount()

O método `componentDidMount()` é um método do ciclo de vida de componentes em React que é chamado após o componente ter sido montado no DOM (Documento Object Model) e está pronto para ser exibido para o usuário.

Este método é executado apenas uma vez durante todo o ciclo de vida do componente e é geralmente usado para realizar ações que só precisam ser feitas uma vez, como buscar dados de um servidor externo ou inicializar bibliotecas de terceiros.

Algumas das ações comuns realizadas no `componentDidMount()` incluem:

- Fazer chamadas a APIs para buscar dados
- Configurar listeners de eventos
- Inicializar bibliotecas de terceiros
- Configurar animações ou transições
- Atualizar o estado do componente com base em informações do DOM

map()

O método map() é uma função de ordem superior (ou seja, uma função que recebe outra função como parâmetro) que existe em arrays em JavaScript. Ele permite transformar um array existente em um novo array com base em uma função de transformação aplicada a cada elemento do array original.

Ele é muito útil em situações em que se deseja transformar um array existente em um novo array com base em alguma lógica de transformação. Ele é muito utilizado em aplicações web para renderizar listas de elementos dinamicamente com base em dados de um array.

BORA CODAR => <https://codesandbox.io/s/apirick-s6z2mi?file=/src/components/Api.js>

Estude o código que fizemos em aula, reproduza e destinche as linhas notando como cada função foi construída e o que está fazendo