

# API E XMLHTTPREQUEST



DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

# AGENDA

- APIs
- Métodos de requisição HTTP
- XMLHttpRequest

- API é uma abreviação para Application Programming Interface, ou, em português, interface de programação de aplicação;
- É um conjunto de definições e protocolos para fazer toda a comunicação com softwares de aplicações;
- Uma API pode ser pensada como um contrato de serviço entre duas aplicações. Esse contrato define como as duas se comunicam usando solicitações e respostas.



- O protocolo HTTP possui e define um conjunto de métodos de requisição responsáveis por indicar a ação a ser executada para um determinado recurso;
- Cada um deles implementa uma semântica diferente dependendo do tipo de requisição pedido
- Ainda é possível passar mais informações a partir de um cabeçalho HTTP.

# MÉTODOS HTTP

- Os principais métodos HTTP são:
  - **GET**: solicita um recurso específico e devem retornar apenas dados;
  - **POST**: é utilizado para enviar um novo dado a um recurso específico (é necessário enviar algo no corpo da requisição);
  - **PUT**: é utilizado para atualizar(enviar) um dado específico (é necessário enviar algo no corpo da requisição);
  - **DELETE**: é utilizado para remover um recurso específico.



# STATUS HTTP

- O status http é sempre retornado ao cliente ao final de uma solicitação para o servidor.
- De acordo com o range pode significar erro ou sucesso.
  - 200 - 226 = Sucesso
  - 400 - 499 = Erro no cliente
  - 500 - 599 = Erro no servidor

Veja os exemplos mais comuns de retorno:



200  
OK



404  
Not Found



500  
Internal Server Error

<https://http.cat/>

# INTERVALO DE AULA

## DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

**Início:** 20:32

**Retorno:** 20:52





# XMLHTTPREQUEST

- É uma das formas usadas para recuperar dados de um servidor de forma assíncrona ;
- Permite que as requisições HTTP sejam feitas via browser;
- O nome sugere que a API é focada em XML, mas elas pode recuperar qualquer tipo de dado, com JSON por exemplo;
- É antiga e hoje não é mais tão amplamente usada devido à sua complexidade;

# XMLHttpRequest

- Para criar uma instância de XMLHttpRequest é usada a sintaxe:

```
const myRequest = new XMLHttpRequest();
```

- Este objeto disponibiliza algumas propriedades para utilizarmos:
  - **readyState**: retorna o [estado](#) de um XMLHttpRequest;
  - **response**: retorna um objeto JavaScript de acordo com o tipo de retorno da api;
  - **status**: retorna o [status code](#) da requisição;

- Este objeto disponibiliza alguns métodos para utilizarmos:
  - **open:** é utilizado para inicializar uma solicitação;
  - **send:** é responsável por enviar a solicitação;
  - **abort:** é utilizado para aborta a solicitação se ela já tiver sido enviada;
- Este objeto disponibiliza alguns eventos para utilizarmos:
  - **onload:** é acionado quando uma transação é concluída com êxito;
  - **onerror:** é acionado quando a solicitação encontrou um erro;
  - **loadend:** é acionado quando uma solicitação é concluída, seja com sucesso ou com erro;
  - **readystatechange:** é acionado sempre que a propriedade readyState é alterada.

# XMLHttpRequest

- Exemplo de implementação com XMLHttpRequest:

```
const xhr = new XMLHttpRequest();
```

chamar a função se a solicitação for um sucesso

```
xhr.onload = () => console.log(xhr.response);
```

```
xhr.onerror = (err) => console.error(err);
```

Recebe por padrão uma função que irá ser executada se a requisição for um sucesso. Nesse caso ele imprime o retorno da chamada xhr

chamar a função se a solicitação der errado

# XMLHTTPREQUEST

- Exemplo de implementação com XMLHttpRequest:

```
const xhr = new XMLHttpRequest();
```

```
xhr.onload = () => console.log(xhr.response);
```

```
xhr.onerror = (err) => console.error(err);
```

```
xhr.open('GET', 'https://viacep.com.br/ws/01001000/json/');
```

Método responsável por receber os parâmetros de requisição.

Os parâmetros em ordem são: o tipo do método (GET, POST, PUT...) e a url da requisição).

# **XMLHTTPREQUEST**

- Exemplo de implementação com XMLHttpRequest:

```
const xhr = new XMLHttpRequest();  
  
xhr.onload = ({ target }) => console.log(target.response);  
  
xhr.onerror = (err) => console.log(err);  
  
xhr.open('GET', 'https://viacep.com.br/ws/01001000/json/');  
  
xhr.send();
```

enviar a solicitação ao servidor

# **XMLHttpRequest**

- Exemplo de implementação com XMLHttpRequest:

```
const xhr = new XMLHttpRequest();
```

criar uma nova instância de XMLHttpRequest

```
xhr.onload = () => console.log(xhr.response);
```

chamar a função se a solicitação for um sucesso

```
xhr.onerror = (err) => console.log(err);
```

chamar a função se a solicitação der errado

```
xhr.open('GET', 'https://viacep.com.br/ws/01001000/json/');
```

abrir uma solicitação GET

```
xhr.send();
```

enviar a solicitação ao servidor

# MATERIAL COMPLEMENTAR

- <https://www.httpstatus.com.br/>
- <https://http.cat/>
- [o que são api?](#)
- <https://viacep.com.br/>
- <https://rapidapi.com/hub>
- [https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Using\\_XMLHttpRequest](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Using_XMLHttpRequest)
- <https://github.com/public-apis/public-apis>
- <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>



## AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





# DEVinHouse

Parcerias para desenvolver a sua carreira

**OBRIGADO!**



<LAB365>