**Fetch API**

**Introdução**

[Fetch API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API) é a mais recente interface para pedidos/respostas a uma rede.

Permite-nos fazer download e upload de recursos como dados [JSON](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON), [XML](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction), Texto ou [Blob](https://developer.mozilla.org/en-US/docs/Web/API/Blob) (imagens). Oferece-nos métodos mais robustos e flexíveis para fazer requisições HTTP por JavaScript em comparação com o antigo método de [XMLHttpRequest (XHR)](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest)

O Principal propósito do [Fetch API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API) é simplificar o processo de pedidos a uma rede em aplicações web, permitindo-nos realizar tarefas comuns como carregar ou enviar dados de a um servidor e lidar com o resultado de uma maneira mais intuitiva e eficiente.

Ao permitir uma forma mais padronizada de trabalhar com requisições de rede, a API promove um código mais limpo e legível tornando mais fácil a gestão de [operações assíncronas](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous) em aplicações web (javascript). Além disso, a API faz uso de tecnologias mais atuais como [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise) e a utilização de sintaxe [async](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function) / [await](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/await), o que simplifica a programação assíncrona e o tratamento de erros.

No geral, o [Fetch API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API) permite que o programador crie aplicações web com uma maior capacidade de resposta, mais dinâmicas e que permite interagir com servidores remotos (ou locais) de uma forma contínua e eficiente. É uma ferramenta essencial para a construção de aplicações web modernas sem depender de bibliotecas ou frameworks externas.

**Exemplo**

Comparação de um simples pedido (GET) usando ambos o [XMLHttpRequest (XHR)](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest) e o [Fetch API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API)

**XHR:**

const xhr = new XMLHttpRequest();

const url = 'data.json';

xhr.onreadystatechange = function() {

if (xhr.readyState === XMLHttpRequest.DONE) {

if (xhr.status === 200) {

console.log(JSON.parse(xhr.responseText));

} else {

console.error('Error:', xhr.status);

}

}

};

xhr.open('GET', url);

xhr.send();

**Fetch:**

try {

const req = await fetch("data.json");

const data = await req.json();

console.log(data);

} catch (error) {

throw new Error(error.message);

}

**Vantagens de Fetch sob XHR**

1. **Simplicidade**
   * A API Fetch fornece uma sintaxe mais moderna e simplificada em comparação com XMLHttpRequest (XHR).
2. **Promise**
   * O resultado da função fetch() é um objeto [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise) permitindo um código mais conciso e legível.
3. **Parse integrado de JSON**
   * O Fetch automaticamente faz o parse de JSON enquanto que em XHR precisamos de explicitamente fazer parse da resposta (JSON.parse(xhr.responseText)).
4. **Suporta mais eficiente para transmissão e manipulação de respostas**
   * O Fetch fornece um melhor suporte para lidar com diferentes tipos de respostas, tais como [Blob](https://developer.mozilla.org/en-US/docs/Web/API/Blob) e [FormData](https://developer.mozilla.org/en-US/docs/Web/API/FormData). Permite também a transmissão de respostas, o que pode ser mais eficiente em termos de memória quando lidamos com grandes quantidades de dados.
5. **Recursos modernos**
   * O Fetch foi projetado para fazer uso de recursos modernos de JavaScript tais como [Promises](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise) e a sintaxe [async](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function)/[await](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/await). Isso facilita a integração com código assíncrono da aplicação web.

**Conclusão**

A [API Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API) oferece uma abordagem mais flexível, moderna e amigável para fazer pedidos em rede nas nossas aplicações web em comparação com o antigo objeto [XHR](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest).