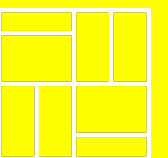


XMLHTTPREQUEST AJAX

É um conjunto de técnicas de desenvolvimento voltado para a web que permite que aplicações trabalhem de modo assíncrono, processando qualquer requisição ao servidor em segundo plano.



XMLHttpRequest

XMLHttpRequest é um objeto que fornece funcionalidade ao cliente para transferir dados entre um cliente e um servidor. Ele fornece uma maneira fácil de recuperar dados de um URL sem ter que fazer uma atualização de página inteira.

Isso permite que uma página da Web atualize apenas uma parte do conteúdo sem interromper o que o usuário esteja fazendo.

XMLHttpRequest é usado constantemente na programação de AJAX.

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

Ajax

AJAX significa Asynchronous JavaScript and XML, ou JavaScript e XML Assíncronos. É um conjunto de técnicas de desenvolvimento web que permite que aplicações trabalhem de modo assíncrono, processando qualquer requisição ao servidor em segundo plano.

O XML (eXtensible Markup Language), como o nome sugere, é uma variação de linguagem markup HTML. Enquanto o HTML é utilizado para apresentar dados, o XML armazena-os e transmite-os.

Métodos de XMLHttpRequest

OPEN(method, URL, async)

O método `open()` inicializa uma nova requisição, ou reinicializa uma requisição já existente.

Os parâmetros principais são:

`method`: método HTTP, como como "GET", "POST", "PUT", "DELETE", etc.

`URL`: uma string representando o URL para enviar a requisição.

`async`: um boolean opcional que indica se realiza a operação de forma assíncrona. O valor default é true.

```
xhr.open('GET', '/exemplo/de/um/url');
```

SEND(body)

O método `send()` envia uma requisição para o servidor.

Aceita um parâmetro opcional que permite especificar o corpo da solicitação; isso é usado principalmente para solicitações como PUT. Se o método de solicitação for GET ou HEAD, o parâmetro body será ignorado e o corpo da solicitação será definido como null.

```
xhr.send();  
xhr.send(null);  
xhr.send('string');
```

Métodos de XMLHttpRequest

ABORT()

O método XMLHttpRequest.abort() aborta a requisição se ela já tiver sido enviada. Quando uma requisição é abortada, o seu readyState é modificado para 0 (Desativado), mas o evento readystatechange não é disparado.

```
var xhr = new XMLHttpRequest(),  
xhr.open("GET","https://developer.mozilla.org/",true);  
xhr.send();  
  
xhr.abort();
```

Exercício

XMLHTTPREQUEST AJAX

- Cria um ficheiro html;
- Cria uma tag script para o código js;
- Faz um request ao endereço
<https://dog.ceo/api/breeds/list/all>
- Verifica na tab Network do browser se o pedido foi feito.

Eventos de XMLHttpRequest

LOAD

O evento load é acionado quando uma transação XMLHttpRequest é concluída com êxito.

```
xhr.addEventListener('load', function(event) {})  
xhr.onload = function(event) {}
```

ERROR

O evento de erro é acionado quando a solicitação encontra um erro.

```
xhr.addEventListener('error', function(event) {})  
xhr.onerror = function(event) {}
```

PROGRESS

O evento de progresso é acionado periodicamente quando uma solicitação recebe mais dados.

```
xhr.addEventListener('progress', function(event) {})  
xhr.onprogress = function(event) {}
```

Eventos de XMLHttpRequest

ABORT

O evento `abort` é acionado quando uma solicitação é abortada, por exemplo, porque o programa chamou `XMLHttpRequest.abort()`.

```
xhr.addEventListener('abort', function(event) {})  
xhr.onabort = function(event) {}
```

READYSTATECHANGE

O evento `onreadystatechange` é acionado toda vez que o `readyState` é alterado. A propriedade `readyState` mantém o status do `XMLHttpRequest`.

```
xhr.addEventListener('readystatechange', function(event) {})  
xhr.onreadystatechange = function(event) {}
```


Propriedades de XMLHttpRequest

READYSTATE

A propriedade XMLHttpRequest.readyState retorna o estado de um XMLHttpRequest. Uma requisição XHR que existe em um dos seguintes estados:

0 - UNSENT	Um cliente foi criado. Mas o método <code>open()</code> não foi chamado ainda.
1 - OPENED	O método <code>open()</code> foi chamado.
2 - HEADERS_RECEIVED	O método <code>send()</code> foi chamado e os cabeçalhos e status estão disponíveis .
3 - LOADING	A carregar e <code>responseText</code> contém os dados parciais.
4 - DONE	Operação concluída.

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4) {
        document.getElementById("demo").innerHTML = 'teste';
    }
}
```

Propriedades de XMLHttpRequest

STATUS

A propriedade `XMLHttpRequest.status` é somente de leitura e retorna o código de status HTTP numérico da resposta do `XMLHttpRequest`.

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (xhttp.status == 200) {
        document.getElementById("demo").innerHTML = xhttp.status;
    }
}
```

RESPONSE

A propriedade `XMLHttpRequest.response` retorna o conteúdo do corpo da resposta.

```
var xhttp = new XMLHttpRequest();
xhttp.onload = function() {
    document.getElementById("demo").innerHTML = xhttp.response;
}
```

Propriedades de XMLHttpRequest

RESPONSETYPE

A propriedade XMLHttpRequest `responseType` é propriedade que especifica o tipo de dados contidos na resposta. Também permite que o autor altere o tipo de resposta. Se uma string vazia for definida como o valor de `responseType`, o valor padrão de `text` será usado.

Tipos de valores mais usados: `text`, `json`

```
var xhttp = new XMLHttpRequest();  
xhttp.responseType = 'json';
```

Adicionar uma Promise a um XMLHttpRequest

```
function myPromiseFunction() {  
    return new Promise(function (resolve, reject) {  
        const request = new XMLHttpRequest();  
        request.open('GET', 'exemplo_url');  
  
        request.onload = function () {  
            resolve(request.response);  
        };  
        request.onerror = function () {  
            reject('erro!');  
        };  
  
        request.send();  
    });  
}
```

Exercício

XMLHTTPREQUEST AJAX

- Cria um ficheiro html;
- Cria uma tag script para o código js;
- Faz um request ao endereço
<https://zoo-animal-api.herokuapp.com/animals/random/3>
- Cria dinamicamente uma galeria de imagens com os animais da resposta. Cada item da galeria deverá conter o nome do animal e uma imagem.
- Aplica neste exercício uma Promise na obtenção de resultados.