

Linguagens de Marcação e Scripts

Prof. Aníbal Cavalcante de Oliveira

UFC - QXD0164 - 2019.2

Agenda - Aula 20

- Realizando chamadas AJAX

Primeira requisição AJAX

Vamos fazer nossa primeira requisição AJAX e obter um texto de resposta.

Passo 1: Criar o documento chamado PrimeiroAjax.html com o código abaixo:

```
<h2>0 Objeto XMLHttpRequest</h2>

<button type="button" onclick="carregarJSON()">Request data</button>

<p id="demo"></p>

<script>
    function carregarJSON() {

    }
</script>
```

Primeira requisição AJAX

Passo 2: Vamos criar nosso objeto **XMLHttpRequest**. Lembrando esse é o objeto JavaScript que fornece funcionalidade ao cliente para transferir dados entre um cliente e um servidor, utilizando requisições e respostas HTTP.

```
<script>
function carregarJSON() {
    var xhttp = new XMLHttpRequest();

}
</script>
```

Primeira requisição AJAX

Passo 3: Vamos utilizar a propriedade **onreadystatechange** do objeto **XMLHttpRequest**. Essa propriedade é chamada sempre que a propriedade **readyState** tem seu valor alterado.

```
<script>
  function carregarJSON() {

    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {

    }
  }
</script>
```

Primeira requisição AJAX

Relembrando que, a propriedade **readyState** retorna o estado de uma requisição. Esses estados podem ser:

Valor	Estado	Descrição
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	Baixando e <code>responseText</code> contem os dados parciais.
4	DONE	Operação concluída.

- * Método **open()** abre uma conexão com o servidor.
- * Método **send()** envia dados para o servidor.
- * **DONE** indica que a operação terminou e temos uma resposta. Isso pode significar que a transferência foi concluída com êxito ou que falhou.

Primeira requisição AJAX

Passo 5: Agora que sabemos que o **readyState** tem como valor **4**. Temos que verificar se o valor da propriedade **status** é igual a **200**, ou seja **SUCESSO**. Ou seja, se servidor retornou uma resposta de sucesso.

```
<script>
  function carregarJSON() {

    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
      if (this.readyState == 4){

      }

    }
  }
</script>
```

Primeira requisição AJAX

Relembrando que, qualquer valor que estiver entre 200 e 299 indica que a requisição foi realizada com sucesso. Por exemplo: 200, 201, 204..

Classe	Semântica
2xx	Indica que a requisição foi processada com sucesso.
3xx	Indica ao cliente uma ação a ser tomada para que a requisição possa ser concluída.
4xx	Indica erro(s) na requisição causado(s) pelo cliente.
5xx	Indica que a requisição não foi concluída devido a erro(s) ocorrido(s) no servidor.

Primeira requisição AJAX

Passo 6: Ajustar o status para obter apenas respostas de sucesso. Ou seja valores entre 200 e 299.

```
<script>

function carregarJSON() {

    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status >= 200 && this.status < 300){

        }
    }
}

</script>
```

Primeira requisição AJAX

Passo 7: Caso o servidor responda com sucesso, podemos obter o valor da resposta com a propriedade `responseText`.

```
<script>

function carregarJSON() {

    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status >= 200 && this.status < 300){
            var resposta = this.responseText;
        }
    }
}

</script>
```

Primeira requisição AJAX

Passo 8: Vamos escrever o conteúdo da resposta, ou seja a propriedade `responseText` dentro do elemento `<p id="demo"></p>`

```
<script>

function carregarJSON() {


    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status >= 200 && this.status < 300){
            var resposta = this.responseText;
            document.getElementById("angel").innerHTML = resposta;
        }
    }
}

</script>
```

il. Primeira requisição AJAX

Passo 9: Agora que sabemos tratar a resposta vamos criar a requisição. Primeiro vamos usar o método **open()** para abrir uma requisição com o verbo **GET**.

```
<script>
function carregarJSON() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status >= 200 && this.status < 300){
            var resposta = this.responseText;
            document.getElementById("demo").innerHTML = resposta;
        }
    }
    xhttp.open("GET", "https://lmsqxd.herokuapp.com/usuarios");
}
</script>
```



Verbo HTTP

URL do serviço na WEB

Primeira requisição AJAX

Passo 10: Por último vamos enviar a requisição com o método `send()`. Como não vamos enviar nenhuma informação no corpo da requisição, colocamos passo o valor `null` para ele.

```
<script>
function carregarJSON() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status >= 200 && this.status < 300) {
            var resposta = this.responseText;
            document.getElementById("demo").innerHTML = resposta;
        }
    }

    xhttp.open("GET", "https://lmsqxd.herokuapp.com/usuarios");
    xhttp.send(null);
}
</script>
```

Primeira requisição AJAX

Passo 11: Agora que obtemos a resposta, uma String JSON, podemos transformá-la em um objeto JavaScript e adicioná-lo na página. Vamos criar um array para receber a String JSON e transformá-la em JS com o comando `JSON.parse()`.

```
<script>
function carregarJSON() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status >= 200 && this.status < 300){
            var resposta = JSON.parse(this.responseText);
            var elemento = document.getElementById("demo");
            elemento.innerHTML += resposta[0].nome + " ";
            elemento.innerHTML += resposta[0].email + " ";
            elemento.innerHTML += resposta[0].nascimento + "<br>";
        }
    }
    xhttp.open("GET", "https://lmsqxd.herokuapp.com/usuarios");
    xhttp.send(null);
}
</script>
```

Primeira requisição AJAX

Exercício 01: Recupere o JSON do slide anterior e mostre todos os dados dos aniversariantes, onde cada pessoa deve estar numa linha. Para isso use uma tabela ou uma lista, e insira os elementos nela.

Primeira requisição AJAX

Exercício 02: Cada contato na agenda de aniversário possui um atributo chamado `_id`, crie um formulário que recebe um valor numérico com o id do usuário e retorna todos os dados do usuário selecionado.

Exemplo de requisição: `https://lmsqxd.herokuapp.com/usuarios/5dcad8b38761cb001755301e`

Resposta:

```
{  "_id": "5cf9496d2b788800171302ad",
    "nome": "Aníbal Cavalcante De Oliveira",
    "email": "hanibal_ce@outlook.com",
    "nascimento": "1981-01-01T00:00:00.001Z",
    "__v": 0
}
```