Informações importantes sobre o AJAX

1-Conceitos Importantes:

1.1-Objetos request e response.

O **'request' ou 'requisição'** é o objeto onde o browser solicita uma página a um servidor web.

O **'response' ou 'resposta'** é o retorno que o servidor envia ao browser do cliente em resposta a requisição feita anteriormente.

A resposta pode ser o que o cliente pediu ou uma mensagem de erro informando qual foi o problema através de um código e mensagem pertinentes ao erro.

1.2-O que são requisições síncronas e assíncronas.

As **requisições síncronas** são aquelas que quando ocorrem o browser aguarda o seu término para poder fazer uma nova requisição. Nesse meio termo o browser ficar 'travado' para o usuário o que é muito ruim.

As **requisições assíncronas** são aquelas que são disparadas quando é solicitado o acesso a um elemento externo ao código da página atual, como uma imagem, um frame ou coisa parecida. Estas requisições tem a vantagem de poderem ser disparadas inúmeras sem que nenhuma atrapalhe a outra (tenha que aguardar) e são tratadas separadamente, ou seja, quando chegar o response do request

2-O que é o AJAX?

A sigla AJAX significa Asynchronous JavaScript e XML e por isso ele usa requisições assíncronas para se comunicar com o servidor.

É um recurso disponível em todos os browsers modernos que expuseram o método httprequest, que antes era exclusivo do browser, para o dom do Javascript. E isso é bem antigo, só pra dar uma idéia, no IE4 foi implementado o AJAX. Isto significa que nada precisa ser instalado para que o AJAX funcione...ele é nativo do browser.

Com isto, através do javascript podemos fazer um 'request' exatamente como se fosse uma página comum mas podemos trocar um innerhtml de uma div, por exemplo, com o 'response' obtido pela chamada. Portanto, o AJAX é muito bom para trocar conteúdos parciais de páginas web sem precisar dar um reload na página.

Portanto o AJAX sempre vai rodar dentro de um script comum javascript.

3-Como funciona o AJAX?

Como foi dito anteriormente, o AJAX faz uma requisição para uma página do servidor e este retorna uma resposta (objeto 'response') e para gerenciar o processo foram incluídos alguns itens que permite verificar erros, se o processo ainda está em execução ou se já terminou.

O processo pode ser dividido em partes como:

3.1-instanciando o objeto httprequest
3.2-conexão com o servidor
3.3-Envio da solicitação
3.4-verificação do andamento da requisição
3.5-requisição terminada e devolvida pelo servidor- Status e erros.

3.1-Instanciando o httprequest no Javascript.

A primeira coisa sempre será informar ao javascript que precisamos instanciar o objeto ajax para termos acesso aos seus recursos e isso é muito fácil de ser feito:

```
let ajax = new XMLHttpRequest();          //instanciando o objeto
```

3.2-Conexão com o servidor

Taí outra coisa bem simples:

```
ajax.open('GET', url)   //enviando a requisição ao servidor
```

O primeiro parâmetro é o método: Os mais comuns são o GET, POST mas existem outros.

Nota: Esses dois primeiros métodos (instanciamento e conexão) são controlados pelo cliente, mas os demais processos são feitos pelo servidor de maneira automática e por este motivo o browser / cliente não tem qualquer controle até o término do processo do lado do servidor.

3.3-Envio da Solicitação.

Neste passo a solicitação é enviada ao servidor.

```
ajax.send()
```

Após o envio da solicitação precisaremos aguardar o término da resposta do servidor e dependendo da resposta, pegar a resposta e exibir.

4-Métodos para verificação do andamento do processo de requisição.

No ajax foi disponibilizado um método que atualiza o status do processo sempre que houver uma mudança nele, o método é:

ajax.onreadystatechange()

E esse método deposita um número correspondente ao estado do processo no DOM, em uma variável para expor o estado da requisição:

ajax.readystate

que pode ter os seguintes valores:

ajax.readystate== 0:request not initialized - Requisição não iniciada

o objeto httprequest foi instanciado (let ajax = new XMLHttpRequest();) mas nada foi feito ainda

ajax.readystate== 1:server connection estabilished - Conexão com servidor estabelecida

o objeto httprequest foi instanciado (let ajax = new XMLHttpRequest();) e foi solicitada a conexão com o servidor: ajax.open('GET', url)

Importante Os dois pontos acima são disparados pelo client mas os demais abaixo são disparados automaticamente pelo server

ajax.readystate== 2:request received:Requisição recebida

O servidor informa que a requisição foi recebida

ajax.readystate== 3:processing request-Processando a requisição

O servidor informa que a requisição foi aceita e que está processando a solicitação

ajax.readystate== 4: request finished and response.ready

O servidor informa que a requisição foi processada e a resposta está pronta para ser utilizada.

Nota: ajax.readystate== 4 Não significa que obteve sucesso, mas sim que foi processada e seja qual for o resultado está disponível.

Quando obtivermos o ajax.readystate== 4 temos a resposta final da requisição do servidor contudo a solicitação pode terminar com sucesso ou com erro. Para verificar qual foi o resultado existe outra variável do dom com esse estado:

ajax.status

O ajax.status é exatamente o request.status code de qualquer página web que pode ser:

1xx : Mensagens de informativas
2xx : Mensagens de sucesso
3xx : Mensagens de redirecionamento
4xx : Respostas de erro do Cliente
5xx : Respostas de erro do Servidor

Portanto se a resposta for 200 a requisição retornou ok, podemos usar o resultado para atualizar nosso objeto da página.

Contudo podemos obter o status 404 que é recurso não encontrado e isso requer outra mensagem ou outro processo:

Exemplo completo:

```
<!DOCTYPE HTML>
<html lang="pt-br">

<!--
    ajax.readystate()   - Ready do Ajax - Indica em que ponto a requisição se
encontra

    0:request not initialized - Requisição não iniciada
        o objeto httprequest foi instanciado ( let ajax = new XMLHttpRequest();)
mas nada foi feito ainda

    1:server connection estabilished - Conexão com servidor estabelecida
        o objeto httprequest foi instanciado ( let ajax = new XMLHttpRequest();)
e foi solicitada a conexão com o
            servidor: ajax.open('GET', url)
        Importante : Os dois pontos acima são disparados pelo client mas os
demais abaixo são disparados
            automaticamente pelo server

    2:request received:Requisição recebida
        O servidor informa que a requisição foi recebida
```

```
    3:processing request-Processando a requisição
        O servidor informa que a requisição foi aceita e que esta processando a
solicitação

    4:request finished and response.ready
        O servidor informa que a requisição foi processada e a resposta esta
pronta para ser utilizada.
        Nota : Não significa que obteve sucesso, mas sim que foi processada e
seja qual for o resultado
            esta disponível.

        Através da função ajax.status podemos testar a resposta que obtivemos do
servidor:
                        No google procure por       request.status code
        1xx : Mensagens de  informativas
        2xx : Mensagens de  sucesso
        3xx : Mensagens de redirecionamento
        4xx : Respostas de erro do Cliente
        5xx : Respostas de erro do Servidor

        a mais comum são as de código 200 ( tudo ok ) e a 404 ( recurso não
localizado )
-->

<!DOCTYPE HTML>
<html lang="pt-br">

       <head>
              <meta charset="UTF-8">
              <title>Requisições assíncronas</title>
              <!-- bootstrap - link cdn -->
              <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">

              <script>
                     function requisitarPagina(url) {
                            document.getElementById('conteudo').innerHTML = ''

                            //incluir o gif de loading na página
                            if(!document.getElementById('loading')) {
                                   let imgLoading = document.createElement('img')
                                   imgLoading.id = 'loading'
                                   imgLoading.src = 'loading.gif'
                                   imgLoading.className = 'rounded mx-auto d-
block'

       document.getElementById('conteudo').appendChild(imgLoading)
                            }

                            let ajax = new XMLHttpRequest();        //instanciando
o objeto

                            //requisição não iniciada, state = 0
                            //console.log(ajax.readyState)

                            ajax.open('GET', url)                   //enviando a
requisição ao servidor

                            //conexão estabelecida com o servidor, state = 1
```

```
                        //console.log(ajax.readyState)


                        //de alguma lógica que fique olhando para o progresso
da req
                        ajax.onreadystatechange = () => {
                              if(ajax.readyState == 4 && ajax.status == 200)
{


      document.getElementById('conteudo').innerHTML = ajax.responseText;

      //document.getElementById('loading').remove()
                              }

                              if(ajax.readyState == 4 && ajax.status == 404)
{


      document.getElementById('conteudo').innerHTML = 'Recurso não localizado'

      //document.getElementById('loading').remove()
                              }
                        }

                        ajax.send()
                        //console.log(ajax)
                  }

            </script>

      </head>

      <body>

            <!-- Static navbar -->
          <nav class="navbar navbar-light bg-light mb-4">
            <div class="container">
              <div class="navbar-brand mb-0 h1">
                    <h3>Requisições síncronas e assíncronas</h3>
              </div>
            </div>
          </nav>


          <div class="container">

             <div class="row mb-2 d-flex justify-content-center">
                    <div class="col-sm-2 mb-2">
                          <a href="#" class="btn btn-primary"
onclick="requisitarPagina('pagina_1_conteudo.html')">Página 1</a>
                    </div>
                    <div class="col-sm-2 mb-2">
                          <a href="#" class="btn btn-primary"
onclick="requisitarPagina('pagina_2_conteudo.html')">Página 2</a>
                    </div>
                    <div class="col-sm-2 mb-2">
                          <a href="#" class="btn btn-primary"
onclick="requisitarPagina('pagina_3_conteudo.html')">Página 3</a>
                    </div>
             </div>
```

```
            <div class="row">
                  <div class="col-md-1"></div>

                  <div class="col-md-10" id="conteudo"></div>

                        <div class="col-md-1"></div>
                  </div>

            </div>
      </body>
</html>
```