

Instituto Federal Goiano - Campus Ceres
Bacharelado em Sistemas de Informação
Prof. Me. Ronneesley Moura Teles

Daniel Moreira Cardoso
Davi Ildeu de Faria
Fernando Maciel
Igor Justino Rodrigues
Paulo Henrique Rodrigues Araujo

O que é AJAX? Como se programa?

Sumário

1	Introdução	2
1.1	Como o AJAX trabalha?	2
2	Aplicação	2
3	Vantagens e Desvantagens	6
3.1	Vantagens	6
3.2	Desvantagens	6
4	Conclusão	6
5	Referências Bibliográficas	7

O que é AJAX? Como se programa?

1 Introdução

O Ajax é um conjunto de tecnologias de desenvolvimento web que quando reunidas permite o desenvolvimento de uma aplicação melhor elaborada, intuitiva, e de melhor desempenho. Ajax Asynchronous JavaScript and XML (JavaScript assíncrono e XML) foi gerado por Jesse James Garret e o nome foi escolhido justamente para facilitar a explicação do modelo de interação diferente entre o navegador e o servidor web.

Asynchronous (assíncrono) é um tipo de comunicação que não ocorre exatamente ao mesmo tempo, não-simultânea. Ajax é o ato de carregar e renderizar uma página usando scripts rodando pelo lado do cliente carregando os dados em plano de fundo sem que haja a necessidade de atualização da página. Ajax não é uma tecnologia, mas sim um conjunto de tecnologias.

1.1 Como o AJAX trabalha?

Enquanto uma aplicação web convencional busca as informações no servidor e retorna ao cliente o AJAX não necessita dessa ação agilizando o processo de carregamento de dados, com o AJAX toda a lógica e processamento é passado para o usuário, quando o usuário faz uma requisição quem busca e trás essas informações é o JavaScript de forma assíncrona evitando o famoso “reload” na tela. O tratamento dos dados, seu formato e exibição fica toda por conta do script que foi carregado inicialmente quando se acessou a página.

O processo inicial de carregamento é mais lento que de uma aplicação comum, pois muitas informações são pré-carregadas. Mas depois, somente os dados são carregados, tornando assim o site mais rápido.

2 Aplicação

A partir de um CRUD feito em PHP, para realizar as interações, realizamos um exemplo para conectar ao banco de dados e inserir sem a necessidade de atualização da página

```
1 CREATE DATABASE IF NOT EXISTS 'ajax';
2
3 USE 'ajax';
4
5 DROP TABLE IF EXISTS 'usuario';
6
7 CREATE TABLE 'usuario' (
8   'id' int(11) NOT NULL AUTOINCREMENT,
9   'nome' varchar(45) DEFAULT NULL,
10  'sexo' varchar(45) DEFAULT NULL,
11  PRIMARY KEY ('id')
12 ) ENGINE=InnoDB AUTOINCREMENT=0 DEFAULT CHARSET=latin1;
```

recursos/codigo/banco.sql

```
1 <?php
2 switch (isset($_GET['opcao']) ? $_GET['opcao'] : "apagar"){
```

```

3     case 'inserir':
4         inserir();
5     break;
6     case 'apagar':
7         apagar();
8     break;
9 }
10
11 function conectar(){
12     $con = new mysqli("localhost", "root", "123456", "ajax");
13     if($con->connect_errno){
14         die("Nao foi possivel conectar, n do erro: " . $con->
15         connect_errno . ", mensagem: " . $con->connect_error);
16     }
17     return $con;
18 }
19
20 function apagar(){
21     $conexao = conectar();
22     if(isset($_GET['id'])){
23         $id = $_GET['id'];
24
25         $conexao->query("delete from usuario where id = $id");
26         $conexao->close();
27     } else {
28
29         $conexao->query("delete from usuario where id > 0");
30         $conexao->close();
31     }
32 }
33
34 function inserir(){
35     if(isset($_POST['nome'])){
36         $nome = $_POST['nome'];
37     }
38     if(isset($_POST['sexo'])){
39         $sexo = $_POST['sexo'];
40     }
41     $conexao = conectar();
42
43     $conexao->query("insert into usuario (nome, sexo) values ('$nome','
44     $sexo')");
45     $conexao->close();
46 }
47 ?>

```

recursos/codigo/CRUD.php

E pela página HTML enviamos os dados:

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="UTF-8">
5         <title>AJAX | Dados</title>
6     </head>
7     <body>
8         <form>
9             <label>Digite seu nome: </label>
10            <input type="text" id="nome" />

```

```

11     <label>Digite seu sexo: </label>
12     <input type="text" id="sexo" />
13     <input id="Inserir" type="button" style="cursor: pointer; text-
14     decoration: underline" value="Inserir" />
15     <label id="status"></label>
16 </form>
17 <script type="text/javascript">
18     (function() {
19         var httpRequest;
20         document.getElementById("Inserir").onclick = function() {
21             var nome = document.getElementById("nome").value;
22             var sexo = document.getElementById("sexo").value;
23             makeRequest('CRUD.php?opcao=inserir ', [nome, sexo]);
24         };
25
26         function makeRequest(url, data) {
27             if (window.XMLHttpRequest) { // Mozilla, Safari, ...
28                 httpRequest = new XMLHttpRequest();
29             } else if (window.ActiveXObject) { // IE
30                 try {
31                     httpRequest = new ActiveXObject("Msxml2.XMLHTTP");
32                 } catch (e) {
33                     try {
34                         httpRequest = new ActiveXObject("Microsoft.XMLHTTP");
35                     } catch (e) {}
36                 }
37             }
38
39             if (!httpRequest) {
40                 alert('Desistindo :( Nao e possivel criar uma instancia
XMLHTTP');
41                 return false;
42             }
43
44             httpRequest.onreadystatechange = alertContents;
45             httpRequest.open('POST', url);
46             httpRequest.setRequestHeader('Content-Type', 'application/x-
www-form-urlencoded ');
47
48             httpRequest.send('nome=' + encodeURIComponent(data[0]) + '&
sexo=' + encodeURIComponent(data[1]));
49         }
50
51         function alertContents() {
52             if (httpRequest.readyState === 4) {
53                 if (httpRequest.status === 200) {
54                     document.getElementById("status").innerHTML = "Registro
Inserido com sucesso!";
55                 } else {
56                     alert('Houve um problema com o pedido. ');
57                 }
58             } else {
59                 document.getElementById("status").innerHTML = "Carregando
...";
60             }
61         }
62     })();
63 </script>
64 </body>
</html>

```

Já no próximo exemplo, informamos o nome e é retornado um alert com as horas, minutos e segundos atuais:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>AJAX | Dados</title>
6   </head>
7   <body>
8     <label>Digite seu nome:
9     <input type="text" id="nome" />
10    </label>
11    <span id="ajaxButton" style="cursor: pointer; text-decoration:
12    underline">
13      Fazer Pedido
14    </span>
15    <script type="text/javascript">
16      (function() {
17        var httpRequest;
18        document.getElementById("ajaxButton").onclick = function() {
19          var nome = document.getElementById("nome").value;
20          makeRequest('dados.php', nome);
21        };
22
23        function makeRequest(url, nome) {
24          if (window.XMLHttpRequest) { // Mozilla, Safari, ...
25            httpRequest = new XMLHttpRequest();
26          } else if (window.ActiveXObject) { // IE
27            try {
28              httpRequest = new ActiveXObject("Msxml2.XMLHTTP");
29            } catch (e) {
30              try {
31                httpRequest = new ActiveXObject("Microsoft.XMLHTTP");
32              } catch (e) {}
33            }
34
35            if (!httpRequest) {
36              alert('Desistindo :( Nao e possivel criar uma instancia
XMLHTTP');
37              return false;
38            }
39
40            httpRequest.onreadystatechange = alertContents;
41            httpRequest.open('POST', url);
42            httpRequest.setRequestHeader('Content-Type', 'application/x-
www-form-urlencoded ');
43
44            httpRequest.send('nome=' + encodeURIComponent(nome));
45          }
46
47          function alertContents() {
48            if (httpRequest.readyState === 4) {
49              if (httpRequest.status === 200) {
50                alert(httpRequest.responseText);
51              } else {
52
```

```

53         alert('Houve um problema com o pedido.');
```

recursos/codigo/requisicao_dados.html

Alert em PHP:

```

1 <?php
2 $name = (isset($_POST['nome'])) ? $_POST['nome'] : 'Nao ha nome';
3
4 $mensagem = "Ola ";
5 $mensagem .= $name . ", agora sao ";
6 $mensagem .= date("H") . " horas " . date("i") . " minutos e " . date
7 ("s") . " segundos ";
8
9 echo $mensagem;
```

recursos/codigo/dados.PHP

3 Vantagens e Desvantagens

3.1 Vantagens

- Uma das principais vantagens do uso do Ajax é o desempenho do site. O Ajax permite buscar dados no servidor e mostrar em apenas uma parte da página assim não se fazendo necessário recarregar toda a página;
- Outra vantagem bastante interessante sobre o Ajax é a compatibilidade. Não importa se a aplicação é em Java, PHP, Python ou outra linguagem de programação. (Sem o uso de frameworks);
- Uma vantagem para o usuário é a interface. Normalmente o uso do Ajax proporciona uma melhor navegabilidade e facilidade de uso para o usuário.

3.2 Desvantagens

- Há uma boa compatibilidade com o Ajax e as linguagens de programação, porém quando usamos frameworks para implementar o Ajax podemos perder essa compatibilidade pois as vezes um framework pode interferir no funcionamento do outro. Como por exemplo a função \$ no JQuery e Prototype.

4 Conclusão

Com o desenvolvimento da pesquisa e implementação dos códigos Ajax foi possível perceber a real necessidade da utilização de tecnologias como esta em um mercado cada vez mais exigente e competitivo como é o nosso. Ainda contribuindo para em ganhos em desempenho e facilidade de uso (para cliente e desenvolvedor).

5 Referências Bibliográficas

[1] <https://imasters.com.br/artigo/10224/ajax/vantagens-e-desvantagens-do-uso-de-ajax-aspectos-praticos?trace=1519021197&source=single>.

[2] https://developer.mozilla.org/pt-BR/docs/AJAX/Getting_Started

[3] <http://www.devfuria.com.br/php/como-funcionam-os-metodos-get-e-post/>