



AJAX

Prof. Me. Hélio Esperidião



O que é?

AJAX é o acrônimo para JavaScript assíncrono + XML.

Não é exatamente uma tecnologia nova, mas um termo empregado em 2005 por Jesse James Garrett para descrever uma nova forma de utilizar em conjunto algumas tecnologias, incluindo HTML ou XHTML, CSS, JavaScript, DOMI, XML, XSLT, e o mais importante:

objeto XMLHttpRequest.

Combinação de tecnologias

As aplicações web que a utilizam são capazes de fazer rapidamente atualizações incrementais para a interface do usuário sem recarregar a página inteira do navegador.

Isso torna a aplicação mais rápida e sensível às ações do usuário.

Histórico de XML

Embora a letra X em AJAX corresponda ao XML, atualmente o JSON é mais utilizado que o XML devido às suas vantagens, como ser parte do JavaScript.

Ambos (JSON e XML) são utilizados para trocar informações entre o cliente e o servidor.

Requisição HTTP

Para fazer uma requisição HTTP ao servidor usando JavaScript, você precisa de uma instância de uma classe que fornece essa funcionalidade.

Essa classe foi originalmente introduzida no Internet Explorer como um objeto ActiveX chamado **XMLHTTP**.

Mozilla, Safari e outros navegadores o seguiram, implementando uma classe XMLHttpRequest que suporta os métodos e propriedades do objeto ActiveX original da Microsoft.



compatível com múltiplos navegadores da classe requerida, você pode fazer o seguinte:

```
var httpRequest;  
if (window.XMLHttpRequest) { // Mozilla, Safari, ...  
    httpRequest = new XMLHttpRequest();  
} else if (window.ActiveXObject) { // IE 8 and older  
    httpRequest = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

Tratar dados recebidos

Para tratar os dados recebidos do servidor você precisa dizer ao objeto requisição HTTP qual função JavaScript irá manipular o processamento da resposta.

Isto é feito definindo a propriedade `onreadystatechange` do objeto para o nome da função JavaScript que deve ser chamada quando o estado da requisição muda, desse jeito:

```
httpRequest.onreadystatechange =  
nameOfTheFunction;
```

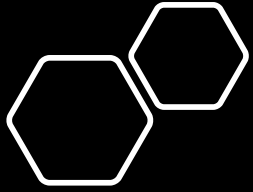
Referência de função



```
httpRequest.onreadystatechange = nameOfTheFunction;
```



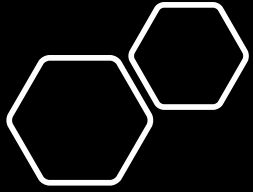
Observe que não existem parênteses depois do nome da função e nenhum parâmetro é passado, porque você está simplesmente atribuindo uma referência à função, ao invés de realmente chamá-la.



Funções anônimas

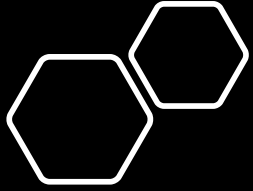
- você pode usar a técnica JavaScript de definir funções dinamicamente (chamadas "funções anônimas") e definir as ações que irão processar de imediato a resposta, dessa forma:

```
httpRequest.onreadystatechange = function(){  
    // processar a resposta do servidor  
};
```



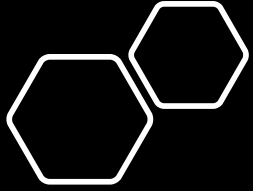
Tipos de requisição: Síncrona

- Quando uma requisição é enviada, o processo remetente é bloqueado até que ocorra uma resposta, ou seja, não é possível enviar novas requisições até que nossa requisição atual seja finalizada, existe sincronismo entre as requisições.



Tipos de requisição: Assíncrona

- Em uma requisição assíncrona, não existe sincronismo entre as requisições, sendo assim, podemos enviar diversas requisições em paralelo, onde cada resposta retorna quando estiver pronta.



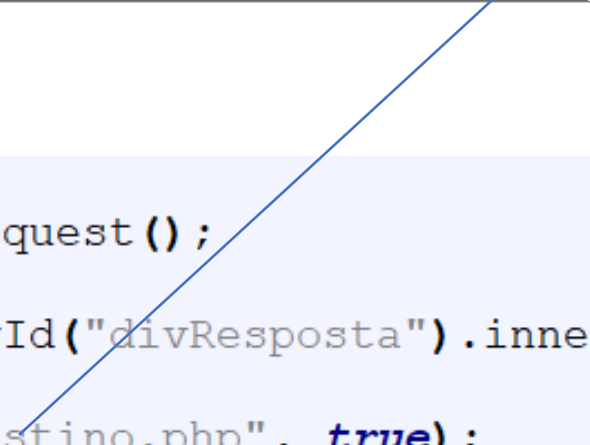
Fazer a requisição

- Em seguida, depois de ter declarado o que vai acontecer assim que receber a resposta, você precisa realmente fazer a requisição. Você precisa chamar os métodos `open()` e `send()` da classe requisição HTTP, dessa forma:
- `httpRequest.open('GET', 'http://www.example.org/some.file', true);`
- `httpRequest.send(null);`

Exemplo Completo

```
<html>
<head>
<script>
  function requisicaoAJAX() {
    const obj = new XMLHttpRequest();
    obj.onload = function() {
      document.getElementById("divResposta").innerHTML = this.responseText;
    }
    obj.open("GET", "paginaDestino.php", true);
    obj.send();
  }
</script>
</head>
<body>
  <input type='button' value ="carregar" onclick='requisicaoAJAX();'>
  <div id='divResposta'> </div>
</body>
</html>
```

```
<?php
    echo "ola mundo";
?>
```



Formulário

```
<html>
<head>
<script>
  function cadastrarCliente() {
    const ajax = new XMLHttpRequest();
    ajax.onload = function() {
      document.getElementById("divResposta").innerHTML = this.responseText;
    }
    ajax.open("POST", "paginaDestino.php");
    ajax.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    var variaveis = "";
    variaveis += "&txtNome=" + document.getElementById('txtNome').value;
    variaveis += "&txtEmail=" + document.getElementById('txtEmail').value;
    variaveis += "&txtSenha=" + document.getElementById('txtSenha').value;
    ajax.send(variaveis);
  }
</script>
</head>
<body>
  <input type='text' id='txtNome'>
  <input type='text' id='txtEmail'>
  <input type='password' id='txtSenha'>

  <input type='button' value="carregar" onclick='cadastrarCliente();'>
  <div id='divResposta'> </div>
</body>
</html>
```

```
<?php
  $nome = $_POST['txtNome'];
  $email = $_POST['txtEmail'];
  $senha = $_POST['txtSenha'];

  echo "RECEBIDO: $nome <br>$email <Br>$senha ";
?>
```

Validar formulário

```
<html>
<head>
<script>
function cadastrarCliente() {
var nome = document.getElementById('txtNome').value;
var email= document.getElementById('txtEmail').value;
var senha = document.getElementById('txtSenha').value;

if(nome==""){
alert("NOME INVÁLIDO");
return;
}
const ajax = new XMLHttpRequest();
ajax.onload = function() {
document.getElementById("divResposta").innerHTML = this.responseText;
}
ajax.open("POST", "paginaDestino.php");
ajax.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
var variaveis = "";
variaveis += "&txtNome=" + nome;
variaveis += "&txtEmail=" + email;
variaveis += "&txtSenha=" + senha;
ajax.send(variaveis);
}
</script>
</head>
<body>
<input type='text' id='txtNome'>
<input type='text' id='txtEmail'>
<input type='password' id='txtSenha'>

<input type='button' value="carregar" onclick='cadastrarCliente();'>
<div id='divResposta'> </div>
</body>
</html>
```

```
<?php
$nome = $_POST['txtNome'];
$email = $_POST['txtEmail'];
$senha = $_POST['txtSenha'];

echo "RECEBIDO: $nome <br>$email <Br>$senha ";
?>
```