# AJAX - Enviar um pedido para um servidor

[❮ Anterior](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_ajax_http.asp&usg=ALkJrhhuyHsspf1BVgj_hZjr3U0e4bf3SA)[Próximo ❯](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_ajax_http_response.asp&usg=ALkJrhhWoiz7tFlioWdLMXfhh9aid8n_SA)

O objeto XMLHttpRequest é usado para trocar dados com um servidor.

## Enviar um pedido para um servidor

Para enviar um pedido para um servidor, usamos os métodos open () e send () do objeto XMLHttpRequest:

xhttp.open("GET", "ajax\_info.txt", true);  
xhttp.send();

|  |  |
| --- | --- |
| **Method** | **Description** |
| open( *method, url, async* ) | Specifies the type of request  *method* : the type of request: GET or POST *url* : the server (file) location *async* : true (asynchronous) or false (synchronous) |
| send() | Sends the request to the server (used for GET) |
| send( *string* ) | Sends the request to the server (used for POST) |

## GET or POST?

GET é mais simples e rápido que POST, e pode ser usado na maioria dos casos.

No entanto, use sempre pedidos POST quando:

* Um arquivo em cache não é uma opção (atualize um arquivo ou banco de dados no servidor).
* Enviando uma grande quantidade de dados para o servidor (o POST não possui limitações de tamanho).
* Enviando entrada do usuário (que pode conter caracteres desconhecidos), o POST é mais robusto e seguro do que o GET.

## Obter solicitações

Um pedido GET simples:

### Exemplo

xhttp.open("GET", "demo\_get.asp", true);  
xhttp.send();

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_ajax_get&usg=ALkJrhh-GwVjjtjvdmUcNY4Fesyk5NTbUA)

No exemplo acima, você pode obter um resultado em cache. Para evitar isso, adicione uma ID exclusiva ao URL:

### Exemplo

xhttp.open("GET", "demo\_get.asp?t=" + Math.random(), true);  
xhttp.send();

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_ajax_get_unique&usg=ALkJrhichxm1QodUfkHAklgNujruduguyw)

Se você deseja enviar informações com o método GET, adicione as informações ao URL:

### Exemplo

xhttp.open("GET", "demo\_get2.asp?fname=Henry&lname=Ford", true);  
xhttp.send();

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_ajax_get2&usg=ALkJrhjIYPbVTi-zZGMNbShhKbWgsykDNA)

## Pedidos POST

Um pedido POST simples:

### Exemplo

xhttp.open("POST", "demo\_post.asp", true);  
xhttp.send();

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_ajax_post&usg=ALkJrhhm27sUZ8gGwcgOiFAKHNL3an92JA)

Para exibir dados como um formulário HTML, adicione um cabeçalho HTTP com setRequestHeader (). Especifique os dados que deseja enviar no método send ():

### Exemplo

xhttp.open("POST", "ajax\_test.asp", true);  
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  
xhttp.send("fname=Henry&lname=Ford");

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_ajax_post2&usg=ALkJrhgLd_QHq5sv36DIiN4_1nakaOvkog)

|  |  |
| --- | --- |
| **Method** | **Description** |
| setRequestHeader( *header, value* ) | Adds HTTP headers to the request  *header* : specifies the header name *value* : specifies the header value |

## A url - Um arquivo em um servidor

O parâmetro url do método open (), é um endereço para um arquivo em um servidor:

xhttp.open("GET", "ajax\_test.asp", true);

O arquivo pode ser qualquer tipo de arquivo, como .txt e .xml, ou arquivos de script do servidor como .asp e .php (que podem executar ações no servidor antes de enviar a resposta de volta).

## Assíncrono - Verdadeiro ou Falso?

Os pedidos de servidor devem ser enviados de forma assíncrona.

O parâmetro async do método open () deve ser definido como verdadeiro:

xhttp.open("GET", "ajax\_test.asp", true);

Ao enviar de forma assíncrona, o JavaScript não precisa esperar pela resposta do servidor, mas pode:

* execute outros scripts enquanto aguarda a resposta do servidor
* lide com a resposta depois que a resposta estiver pronta

## A propriedade onreadystatechange

Com o objeto XMLHttpRequest, você pode definir uma função a ser executada quando a solicitação receber uma resposta.

A função é definida na propriedade **onreadystatechange** do objeto XMLHttpResponse:

### Exemplo

xhttp.onreadystatechange = function() {  
  if (this.readyState == 4 && this.status == 200) {  
    document.getElementById("demo").innerHTML = this.responseText;  
  }  
};  
xhttp.open("GET", "ajax\_info.txt", true);  
xhttp.send();

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_ajax_first&usg=ALkJrhgWcDAaiZTVHYie-YJEgMe-muERBw)

Você aprenderá mais sobre onreadystatechange em um capítulo posterior.

## Solicitação síncrona

Para executar uma solicitação síncrona, altere o terceiro parâmetro no método open () para false:

xhttp.open("GET", "ajax\_info.txt", false);

Às vezes, async = false são usados ​​para testes rápidos. Você também encontrará solicitações síncronas no código JavaScript mais antigo.

Uma vez que o código aguardará a conclusão do servidor, não há necessidade de uma função onreadystatechange:

### Exemplo

xhttp.open("GET", "ajax\_info.txt", false);  
xhttp.send();  
document.getElementById("demo").innerHTML = xhttp.responseText;

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_ajax_asyncfalse&usg=ALkJrhgV2X9XoyqJ0GIkqaDHa1o6f7mn-w)

Synchronous XMLHttpRequest (async = false) não é recomendado porque o JavaScript irá parar de executar até que a resposta do servidor esteja pronta. Se o servidor estiver ocupado ou lento, o aplicativo irá pendurar ou parar.

Synchronous XMLHttpRequest está em processo de remoção do padrão web, mas esse processo pode levar muitos anos.

As ferramentas de desenvolvimento modernas são encorajadas a alertar sobre o uso de solicitações síncronas e podem lançar uma exceção InvalidAccessError quando ocorre.