## **向服务器发送请求**

如需将请求发送到服务器，我们使用 XMLHttpRequest 对象的 open() 和 send() 方法：

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

|  |  |
| --- | --- |
| **方法** | **描述** |
| open(*method*,*url*,*async*) | 规定请求的类型、URL 以及是否异步处理请求。  *method*：请求的类型；GET 或 POST  *url*：文件在服务器上的位置  *async*：true（异步）或 false（同步） |
| send(*string*) | 将请求发送到服务器。  *string*：仅用于 POST 请求 |

## **GET 还是 POST？**

与 POST 相比，GET 更简单也更快，并且在大部分情况下都能用。

然而，在以下情况中，请使用 POST 请求：

无法使用缓存文件（更新服务器上的文件或数据库）

向服务器发送大量数据（POST 没有数据量限制）

发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

一个简单的 GET 请求：

xmlhttp.open("GET","demo\_get.html",true);  
xmlhttp.send();

## **POST 请求**

一个简单 POST 请求：

xmlhttp.open("POST","demo\_post.html",true);  
xmlhttp.send();

如果需要像 HTML 表单那样 POST 数据，请使用 setRequestHeader() 来添加 HTTP 头。然后在 send() 方法中规定您希望发送的数据：

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

|  |  |
| --- | --- |
| **方法** | **描述** |
| setRequestHeader(*header,value*) | 向请求添加 HTTP 头。  *header*: 规定头的名称  *value*: 规定头的值 |

## **url - 服务器上的文件**

open() 方法的 *url* 参数是服务器上文件的地址：

xmlhttp.open("GET","ajax\_test.html",true);

该文件可以是任何类型的文件，比如 .txt 和 .xml，或者服务器脚本文件，比如 .asp 和 .php （在传回响应之前，能够在服务器上执行任务）。

## **Async=true**

当使用 async=true 时，可以规定在响应处于 onreadystatechange 事件中的就绪状态时执行的函数：

## **实例**

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

## **Async = false**

如需使用 async=false，请将 open() 方法中的第三个参数改为 false：

xmlhttp.open(&quot;GET&quot;,&quot;test1.txt&quot;,false);