XMLHttpRequest 用于在后台与服务器交换数据。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

所有现代浏览器（IE7+、Firefox、Chrome、Safari 以及 Opera）均内建 XMLHttpRequest 对象。

创建 XMLHttpRequest 对象的语法：

*variable*=new XMLHttpRequest();

老版本的 Internet Explorer （IE5 和 IE6）使用 ActiveX 对象：

*variable*=new ActiveXObject("Microsoft.XMLHTTP");

为了应对所有的现代浏览器，包括 IE5 和 IE6，请检查浏览器是否支持 XMLHttpRequest 对象。如果支持，则创建 XMLHttpRequest 对象。如果不支持，则创建 ActiveXObject ：:

**实例**

var xmlhttp;  
if (window.XMLHttpRequest)  
  {// code for IE7+, Firefox, Chrome, Opera, Safari  
  xmlhttp=new XMLHttpRequest();  
  }  
else  
  {// code for IE6, IE5  
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  
  }

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 |

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

**GET 请求**

一个简单的 GET 请求：

**实例**

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

在上面的例子中，您可能得到的是缓存的结果。

为了避免这种情况，请向 URL 添加一个唯一的 ID：

**实例**

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

如果您希望通过 GET 方法发送信息，请向 URL 添加信息：

**实例**

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

**POST 请求**

一个简单 POST 请求：

**实例**

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：

**实例**

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

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

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）。

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

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();

**onreadystatechange 事件**

当请求被发送到服务器时，我们需要执行一些基于响应的任务。

每当 readyState 改变时，就会触发 onreadystatechange 事件。

readyState 属性存有 XMLHttpRequest 的状态信息。

下面是 XMLHttpRequest 对象的三个重要的属性：

|  |  |
| --- | --- |
| **属性** | **描述** |
| onreadystatechange | 存储函数（或函数名），每当 readyState 属性改变时，就会调用该函数。 |
| readyState | 存有 XMLHttpRequest 的状态。从 0 到 4 发生变化。   * 0: 请求未初始化 * 1: 服务器连接已建立 * 2: 请求已接收 * 3: 请求处理中 * 4: 请求已完成，且响应已就绪 |
| status | 200: "OK" 404: 未找到页面 |

在 onreadystatechange 事件中，我们规定当服务器响应已做好被处理的准备时所执行的任务。

当 readyState 等于 4 且状态为 200 时，表示响应已就绪：

**注意：** onreadystatechange 事件被触发 5 次（0 - 4），对应着 readyState 的每个变化。

如需获得来自服务器的响应，请使用 XMLHttpRequest 对象的 responseText 或 responseXML 属性。

|  |  |
| --- | --- |
| **属性** | **描述** |
| responseText | 获得字符串形式的响应数据。 |
| responseXML | 获得 XML 形式的响应数据。 |

如果来自服务器的响应并非 XML，请使用 responseText 属性。

responseText 属性返回字符串形式的响应，因此您可以这样使用：

**实例**

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

如果来自服务器的响应是 XML，而且需要作为 XML 对象进行解析，请使用 responseXML 属性：

**实例**

请求 [cd\_catalog.xml](http://www.runoob.com/try/demo_source/cd_catalog.xml) 文件，并解析响应：

xmlDoc=xmlhttp.responseXML;  
txt="";  
x=xmlDoc.getElementsByTagName("ARTIST");  
for (i=0;i<x.length;i++)  
  {  
  txt=txt + x[i].childNodes[0].nodeValue + "<br>";  
  }  
document.getElementById("myDiv").innerHTML=txt;

回调函数是一种以参数形式传递给另一个函数的函数。

如果您的网站上存在多个 AJAX 任务，那么您应该为创建 XMLHttpRequest 对象编写一个标准的函数，并为每个 AJAX 任务调用该函数。

该函数调用应该包含 URL 以及发生 onreadystatechange 事件时执行的任务（每次调用可能不尽相同）：

<body>

<div id="myDiv"><h2>使用 AJAX 修改文本内容</h2></div>

<button type="button" onclick="myFunction()">修改内容</button>

</body>

<script>

var xmlhttp;

function loadXMLDoc(url,cfunc)

{

if (window.XMLHttpRequest)

{// IE7+, Firefox, Chrome, Opera, Safari 代码

xmlhttp=new XMLHttpRequest();

}

else

{// IE6, IE5 代码

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=cfunc;

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

function myFunction()

{

loadXMLDoc("/try/ajax/ajax\_info.txt",function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

});

}

</script>

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)

我们不推荐使用 async=false，但是对于一些小型的请求，也是可以的。

请记住，JavaScript 会等到服务器响应就绪才继续执行。如果服务器繁忙或缓慢，应用程序会挂起或停止。

**注意：**当您使用 async=false 时，请不要编写 onreadystatechange 函数 - 把代码放到 send() 语句后面即可：

**实例**

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

下面的例子将为您演示当用户在输入框中键入字符时，网页如何与 web 服务器进行通信： 请在下面的输入框中键入字母（A - Z）,suggest会显示值，该值从后台数据查找首字母为输入字母的词语：

**Start typing a name in the input field below:**

窗体顶端

First name: 

窗体底端

Suggestions:

<body>

<h3>Start typing a name in the input field below:</h3>

<form action="">

First name: <input type="text" id="txt1" onkeyup="showHint(this.value)" />

</form>

<p>Suggestions: <span id="txtHint"></span></p>

</body>

<script>

function showHint(str)

{

var xmlhttp;

if (str.length==0)

{

document.getElementById("txtHint").innerHTML="";

return;

}

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("txtHint").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET","/try/ajax/gethint.php?q="+str,true);

xmlhttp.send();

}

</script>

如果输入框为空 (str.length==0)，则该函数清空 txtHint 占位符的内容，并退出函数。

如果输入框不为空，showHint() 函数执行以下任务：

创建 XMLHttpRequest 对象

当服务器响应就绪时执行函数

把请求发送到服务器上的文件

请注意我们向 URL 添加了一个参数 q （带有输入框的内容）

下面，我们创建了两个版本的服务器文件，一个用 ASP 编写，另一个用 PHP 编写。

"gethint.asp" 中的源代码会检查一个名字数组，然后向浏览器返回相应的名字：

<%  
response.expires=-1  
dim a(30)  
'Fill up array with names  
a(1)="Anna"  
a(2)="Brittany"  
a(3)="Cinderella"  
a(4)="Diana"  
a(5)="Eva"  
a(6)="Fiona"  
a(7)="Gunda"  
a(8)="Hege"  
a(9)="Inga"  
a(10)="Johanna"  
a(11)="Kitty"  
a(12)="Linda"  
a(13)="Nina"  
a(14)="Ophelia"  
a(15)="Petunia"  
a(16)="Amanda"  
a(17)="Raquel"  
a(18)="Cindy"  
a(19)="Doris"  
a(20)="Eve"  
a(21)="Evita"  
a(22)="Sunniva"  
a(23)="Tove"  
a(24)="Unni"  
a(25)="Violet"  
a(26)="Liza"  
a(27)="Elizabeth"  
a(28)="Ellen"  
a(29)="Wenche"  
a(30)="Vicky"  
  
'get the q parameter from URL  
q=ucase(request.querystring("q"))  
  
'lookup all hints from array if length of q>0  
if len(q)>0 then  
  hint=""  
  for i=1 to 30  
    if q=ucase(mid(a(i),1,len(q))) then  
      if hint="" then  
        hint=a(i)  
      else  
        hint=hint & " , " & a(i)  
      end if  
    end if  
  next  
end if  
  
'Output "no suggestion" if no hint were found  
'or output the correct values  
if hint="" then  
  response.write("no suggestion")  
else  
  response.write(hint)  
end if  
%>

下面的代码用 PHP 编写，与上面的 ASP 代码作用是一样的。

<?php  
// Fill up array with names  
$a[]="Anna";  
$a[]="Brittany";  
$a[]="Cinderella";  
$a[]="Diana";  
$a[]="Eva";  
$a[]="Fiona";  
$a[]="Gunda";  
$a[]="Hege";  
$a[]="Inga";  
$a[]="Johanna";  
$a[]="Kitty";  
$a[]="Linda";  
$a[]="Nina";  
$a[]="Ophelia";  
$a[]="Petunia";  
$a[]="Amanda";  
$a[]="Raquel";  
$a[]="Cindy";  
$a[]="Doris";  
$a[]="Eve";  
$a[]="Evita";  
$a[]="Sunniva";  
$a[]="Tove";  
$a[]="Unni";  
$a[]="Violet";  
$a[]="Liza";  
$a[]="Elizabeth";  
$a[]="Ellen";  
$a[]="Wenche";  
$a[]="Vicky";  
  
//get the q parameter from URL  
$q=$\_GET["q"];  
  
//lookup all hints from array if length of q>0  
if (strlen($q) > 0)  
  {  
  $hint="";  
  for($i=0; $i<count($a); $i++)  
    {  
    if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))  
      {  
      if ($hint=="")  
        {  
        $hint=$a[$i];  
        }  
      else  
        {  
        $hint=$hint." , ".$a[$i];  
        }  
      }  
    }  
  }  
  
// Set output to "no suggestion" if no hint were found  
// or to the correct values  
if ($hint == "")  
  {  
  $response="no suggestion";  
  }  
else  
  {  
  $response=$hint;  
  }  
  
//output the response  
echo $response;  
?>

下面的例子将演示网页如何通过 AJAX 从数据库读取信息： 请在下面的下拉列表中选择一个客户：

**Example**

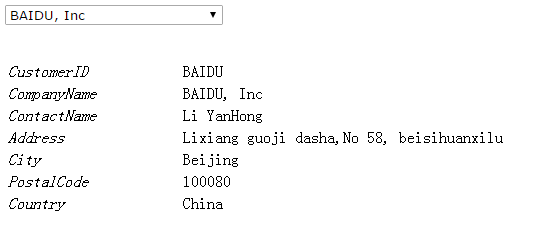
窗体顶端



窗体底端

Customer info will be listed here...

结果



<body>

<form action="">

<select name="customers" onchange="showCustomer(this.value)" style="font-family:Verdana, Arial, Helvetica, sans-serif;">

<option value="APPLE">Apple Computer, Inc.</option>

<option value="BAIDU ">BAIDU, Inc</option>

<option value="Canon">Canon USA, Inc.</option>

<option value="Google">Google, Inc.</option>

<option value="Nokia">Nokia Corporation</option>

<option value="SONY">Sony Corporation of America</option>

</select>

</form>

<br>

<div id="txtHint">Customer info will be listed here...</div>

</body>

<script>

function showCustomer(str)

{

var xmlhttp;

if (str=="")

{

document.getElementById("txtHint").innerHTML="";

return;

}

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("txtHint").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET","/try/ajax/getcustomer.php?q="+str,true);

xmlhttp.send();

}

</script>

由上面的 JavaScript 调用的服务器页面是 PHP 文件，名为 "getcustomer.php"。

用 PHP 编写服务器文件也很容易，或者用其他服务器语言。请看[用 PHP 编写的相应的例子](http://www.runoob.com/php/php-ajax-database.html" \o "PHP 和 AJAX MySQL 数据库实例)。

"getcustomer.php" 中的源代码负责对数据库进行查询，然后用 HTML 表格返回结果：

<%  
response.expires=-1  
sql="SELECT \* FROM CUSTOMERS WHERE CUSTOMERID="  
sql=sql & "'" & request.querystring("q") & "'"  
  
set conn=Server.CreateObject("ADODB.Connection")  
conn.Provider="Microsoft.Jet.OLEDB.4.0"  
conn.Open(Server.Mappath("/db/northwind.mdb"))  
set rs=Server.CreateObject("ADODB.recordset")  
rs.Open sql,conn  
  
response.write("<table>")  
do until rs.EOF  
  for each x in rs.Fields  
    response.write("<tr><td><b>" & x.name & "</b></td>")  
    response.write("<td>" & x.value & "</td></tr>")  
  next  
  rs.MoveNext  
loop  
response.write("</table>")  
%>

下面的例子将演示网页如何使用 AJAX 来读取来自 XML 文件的信息：

<body>

<div id="txtCDInfo">

<button onclick="loadXMLDoc('cd\_catalog.xml')">Get CD info</button>

</div>

</body>

</html>

<script>

function loadXMLDoc(url)

{

var xmlhttp;

var txt,x,xx,i;

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

txt="<table border='1'><tr><th>Title</th><th>Artist</th></tr>";

x=xmlhttp.responseXML.documentElement.getElementsByTagName("CD");

for (i=0;i<x.length;i++)

{

txt=txt + "<tr>";

xx=x[i].getElementsByTagName("TITLE");

{

try

{

txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

}

catch (er)

{

txt=txt + "<td>&nbsp;</td>";

}

}

xx=x[i].getElementsByTagName("ARTIST");

{

try

{

txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";

}

catch (er)

{

txt=txt + "<td>&nbsp;</td>";

}

}

txt=txt + "</tr>";

}

txt=txt + "</table>";

document.getElementById('txtCDInfo').innerHTML=txt;

}

}

xmlhttp.open("GET",url,true);

xmlhttp.send();

}

</script>

**AJAX 服务器页面**

上面这个例子中使用的服务器页面实际上是一个名为 "[cd\_catalog.xml](http://www.runoob.com/try/demo_source/cd_catalog.xml)" XML 文件。

<!-- Edited by XMLSpy® -->

<CATALOG>

<CD>

<TITLE>Empire Burlesque</TITLE>

<ARTIST>Bob Dylan</ARTIST>

<COUNTRY>USA</COUNTRY>

<COMPANY>Columbia</COMPANY>

<PRICE>10.90</PRICE>

<YEAR>1985</YEAR>

</CD>

<CD>

<TITLE>Greatest Hits</TITLE>

<ARTIST>Dolly Parton</ARTIST>

<COUNTRY>USA</COUNTRY>

<COMPANY>RCA</COMPANY>

<PRICE>9.90</PRICE>

<YEAR>1982</YEAR>

</CD>

。。。多个<CD></CD>

</CATALOG>