

## XmlHttp是什么？

最通用的定义为：XmlHttp是一套可以在Javascript、VbScript、Jscript等脚本语言中通过http协议传送或从接收XML及其他数据的一套API。XmlHttp最大的用处是可以更新网页的部分内容而不需要刷新整个页面。

来自MSDN的解释：XmlHttp提供客户端同http服务器通讯的协议。客户端可以通过XmlHttp对象(MSXML2.XMLHTTP.3.0)向http服务器发送请求并使用微软XML文档对象模型Microsoft® XML Document Object Model (DOM)处理回应。

现在的绝对多数浏览器都增加了对XmlHttp的支持，IE中使用ActiveXObject方式创建XmlHttp对象，其他浏览器如：Firefox、Opera等通过window.XMLHttpRequest来创建xmlhttp对象。

## XmlHttp对象参考：

### 属性：

onreadystatechange*	指定当readyState属性改变时的事件处理句柄。只写
readyState	返回当前请求的状态，只读。
responseBody	将回应信息正文以unsigned byte数组形式返回。只读
responseStream	以Ado Stream对象的形式返回响应信息。只读
responseText	将响应信息作为字符串返回。只读
responseXML	将响应信息格式化为Xml Document对象并返回，只读
status	返回当前请求的http状态码。只读
statusText	返回当前请求的响应行状态，只读

\* 表示此属性是W3C文档对象模型的扩展。

### 方法：

abort	取消当前请求
getAllResponseHeaders	获取响应的所有http头
getResponseHeader	从响应信息中获取指定的http头
open	创建一个新的http请求，并指定此请求的方法、URL以及验证信息(用户名/密码)
send	发送请求到http服务器并接收回应
setRequestHeader	单独指定请求的某个http头

### 事件：

无

## XMLHttpRequest

---

### - XMLHTTP对象参

考

[XMLHttpRequest 对象](#)

[XMLHttpRequest 成员](#)

[onreadystatechange](#)

[readyState](#)

[responseBody](#)

[responseStream](#)

[responseText](#)

[responseXML](#)

[status](#)

[statusText](#)

[abort](#)

[getAllResponseHeaders](#)

[getResponseHeader](#)

[open](#)

[send](#)

[setRequestHeader](#)

## **onreadystatechange**

指定当readyState属性改变时的事件处理句柄

## 语法

```
oXMLHttpRequest.onreadystatechange = funcMyHandler;
```

## Example

如下的例子演示当XMLHttpRequest对象的readyState属性改变时调用HandleStateChange函数，当数据接收完毕后（readystate == 4）此页面上的一个按钮将被激活

```
var xmlhttp=null;
function PostOrder(xmldoc)
{
    var xmlhttp = new ActiveXObject("Msxml2.XMLHTTP.5.0");
    xmlhttp.Open("POST", "http://myserver/orders/processorder.asp", f
    xmlhttp.onreadystatechange= HandleStateChange;
    xmlhttp.Send(xmldoc);
    myButton.disabled = true;
}
function HandleStateChange()
{
    if (xmlhttp.readyState == 4)
    {
        myButton.disabled = false;
        alert("Result = " + xmlhttp.responseXML.xml);
    }
}
```

## 备注

此属性只写，为W3C文档对象模型的扩展.

## 参考

readyState 属性

## XMLHttpRequest

---

### - XMLHTTP对象参

考

[XMLHttpRequest 对象](#)

[XMLHttpRequest 成员](#)

[onreadystatechange](#)

[readyState](#)

[responseBody](#)

[responseStream](#)

[responseText](#)

[responseXML](#)

[status](#)

[statusText](#)

[abort](#)

[getAllResponseHeaders](#)

[getResponseHeader](#)

[open](#)

[send](#)

[setRequestHeader](#)



## **responseText**

将响应信息作为字符串返回

## 语法

```
strValue = oXMLHttpRequest.responseText;
```

## Example

```
var xmlhttp = new ActiveXObject("Msxml2.XMLHTTP.3.0");  
xmlhttp.open("GET", "http://localhost/books.xml", false);  
xmlhttp.send();  
alert(xmlhttp.responseText);
```

## 备注

变量，此属性只读，将响应信息作为字符串返回。  
XMLHTTP尝试将响应信息解码为Unicode字符串，XMLHTTP默认将响应数据的编码定为UTF-8，如果服务器返回的数据带BOM(byte-order mark)，XMLHTTP可以解码任何UCS-2 (big or little endian)或者UCS-4 数据。注意，如果服务器返回的是xml文档，此属性并不处理xml文档中的编码声明。你需要使用[responseXML](#)来处理。

## 参考

[responseBody 属性](#)

[responseText 属性](#)

[responseXML 属性](#)

## XMLHttpRequest

---

### - XMLHTTP对象参

考

[XMLHttpRequest 对象](#)

[XMLHttpRequest 成员](#)

[onreadystatechange](#)

[readyState](#)

[responseBody](#)

[responseStream](#)

[responseText](#)

[responseXML](#)

[status](#)

[statusText](#)

[abort](#)

[getAllResponseHeaders](#)

[getResponseHeader](#)

[open](#)

[send](#)

[setRequestHeader](#)

## **status**

返回当前请求的http状态码

## 语法

```
lValue = oXMLHttpRequest.status;
```



## Example

```
var xmlhttp = new ActiveXObject("Msxml2.XMLHTTP.3.0");  
xmlhttp.open("GET", "http://localhost/books.xml", false);  
xmlhttp.send();  
alert(xmlhttp.status);
```

## 返回值

长整形标准http状态码，定义如下：

<b>Number</b>	<b>Description</b>
100	Continue
101	Switching protocols
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found

303	See Other
304	Not Modified
305	Use Proxy
307	Temporary Redirect
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout

409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
416	Requested Range Not Suitable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable

504	Gateway Timeout
505	HTTP Version Not Supported

## 备注

长整形，此属性只读，返回当前请求的http状态码,此属性仅当数据发送并接收完毕后才可获取。

## 参考

[statusText 属性send 方法](#)

## XMLHttpRequest

---

### - XMLHTTP对象参

考

[XMLHttpRequest 对象](#)

[XMLHttpRequest 成员](#)

[onreadystatechange](#)

[readyState](#)

[responseBody](#)

[responseStream](#)

[responseText](#)

[responseXML](#)

[status](#)

[statusText](#)

[abort](#)

[getAllResponseHeaders](#)

[getResponseHeader](#)

[open](#)

[send](#)

[setRequestHeader](#)



**abort**

取消当前请求

## 语法

```
XMLHttpRequest.abort();
```

## 备注

调用此方法后，当前请求返回UNINITIALIZED 状态。

## 参考

[readyState 属性](#)  
[open 方法](#)

## XMLHttpRequest

---

### - XMLHTTP对象参

考

[XMLHttpRequest 对象](#)

[XMLHttpRequest 成员](#)

[onreadystatechange](#)

[readyState](#)

[responseBody](#)

[responseStream](#)

[responseText](#)

[responseXML](#)

[status](#)

[statusText](#)

[abort](#)

[getAllResponseHeaders](#)

[getResponseHeader](#)

[open](#)

[send](#)

[setRequestHeader](#)

## open

创建一个新的http请求，并指定此请求的方法、URL以及验证信息

## 语法

```
oXMLHttpRequest.open(bstrMethod, bstrUrl, varAsync, bstrUser, bstrP
```

## 参数

*bstrMethod* http方法，例如：POST、GET、PUT及PROPFIND。大小写不敏感。

*bstrUrl*

请求的URL地址，可以为绝对地址也可以为相对地址。

*varAsync*[可选]

布尔型，指定此请求是否为异步方式，默认为true。如果为真，当状态改变时会调用onreadystatechange属性指定的回调函数。

*bstrUser*[可选]

如果服务器需要验证，此处指定用户名，如果未指定，当服务器需要验证时，会弹出验证窗口。

*bstrPassword*[可选]

验证信息中的密码部分，如果用户名为空，则此值将被忽略。



## Example

下面的例子演示从服务器请求book.xml,并显示其中的book字段。

```
var xmlhttp = new ActiveXObject("Msxml2.XMLHTTP.3.0");
xmlhttp.open("GET", "http://localhost/books.xml", false);
xmlhttp.send();
var book = xmlhttp.responseXML.selectSingleNode("//book[@id='bk101']");
alert(book.xml);
```

## 备注

调用此方法后，可以调用send方法向服务器发送数据。

## 参考

[abort 方法](#)

[onreadystatechange 属性](#)

[XMLHttpRequest 对象](#)

## XMLHttpRequest

---

### - XMLHTTP对象参

考

[XMLHttpRequest 对象](#)

[XMLHttpRequest 成员](#)

[onreadystatechange](#)

[readyState](#)

[responseBody](#)

[responseStream](#)

[responseText](#)

[responseXML](#)

[status](#)

[statusText](#)

[abort](#)

[getAllResponseHeaders](#)

[getResponseHeader](#)

[open](#)

[send](#)

[setRequestHeader](#)

**send**

发送请求到http服务器并接收回应

## 语法

```
oXMLHttpRequest.send(varBody);
```

## 参数

*varBody* 欲通过此请求发送的数据。

## Example

```
xmlhttp = new ActiveXObject("Msxml2.XMLHTTP.3.0");  
xmlhttp.open("GET", "http://localhost/sample.xml", false);  
xmlhttp.send();  
alert(xmlhttp.responseXML.xml);
```



## 备注

此方法的同步或异步方式取决于open方法中的bAsync参数，如果bAsync == False，此方法将会等待请求完成或者超时时才会返回，如果bAsync == True，此方法将立即返回。

This method takes one optional parameter, which is the requestBody to use. The acceptable VARIANT input types are BSTR, SAFEARRAY of UI1 (unsigned bytes), IDispatch to an XML Document Object Model (DOM) object, and IStream \*. You can use only chunked encoding (for sending) when sending IStream \* input types. The component automatically sets the Content-Length header for all but IStream \* input types.

如果发送的数据为BSTR，则回应被编码为utf-8, 必须在适当位置设置一个包含charset的文档类型头。

If the input type is a SAFEARRAY of UI1, the response is sent as is without additional encoding. The caller must set a Content-Type header with the appropriate content type.

如果发送的数据为XML DOM object，则回应将被编码为在xml文档中声明的编码，如果在xml文档中没有声明编码，则使用默认的UTF-8。

If the input type is an IStream \*, the response is sent as is without additional encoding. The caller must set a Content-Type header with the appropriate content type.

## 参考

[open 方法](#)

[XMLHttpRequest 对象](#)

## XmlHttp是什么？

最通用的定义为：XmlHttp是一套可以在Javascript、VbScript、Jscript等脚本语言中通过http协议传送或从接收XML及其他数据的一套API。XmlHttp最大的用处是可以更新网页的部分内容而不需要刷新整个页面。

来自MSDN的解释：XmlHttp提供客户端同http服务器通讯的协议。客户端可以通过XmlHttp对象(MSXML2.XMLHTTP.3.0)向http服务器发送请求并使用微软XML文档对象模型Microsoft® XML Document Object Model (DOM)处理回应。

现在的绝对多数浏览器都增加了对XmlHttp的支持，IE中使用ActiveXObject方式创建XmlHttp对象，其他浏览器如：Firefox、Opera等通过window.XMLHttpRequest来创建xmlhttp对象。

## XmlHttp对象参考：

### 属性：

onreadystatechange*	指定当readyState属性改变时的事件处理句柄。只写
readyState	返回当前请求的状态，只读。
responseBody	将回应信息正文以unsigned byte数组形式返回。只读
responseStream	以Ado Stream对象的形式返回响应信息。只读
responseText	将响应信息作为字符串返回。只读
responseXML	将响应信息格式化为Xml Document对象并返回，只读
status	返回当前请求的http状态码。只读
statusText	返回当前请求的响应行状态，只读

\* 表示此属性是W3C文档对象模型的扩展。

### 方法：

abort	取消当前请求
getAllResponseHeaders	获取响应的所有http头
getResponseHeader	从响应信息中获取指定的http头
open	创建一个新的http请求，并指定此请求的方法、URL以及验证信息(用户名/密码)
send	发送请求到http服务器并接收回应
setRequestHeader	单独指定请求的某个http头

### 事件：

无

- XMLHTTP对象参

考

[XMLHttpRequest 对象](#)

[XMLHttpRequest 成员](#)

[onreadystatechange](#)

[readyState](#)

[responseBody](#)

[responseStream](#)

[responseText](#)

[responseXML](#)

[status](#)

[statusText](#)

[abort](#)

[getAllResponseHeaders](#)

[getResponseHeader](#)

[open](#)

[send](#)

[setRequestHeader](#)

## XMLHttpRequest成员

### 属性

<code>onreadystatechange*</code>	指定当readyState属性改变时的事件处理句柄。只写
<code>readyState</code>	返回当前请求的状态，只读。
<code>responseBody</code>	将回应信息正文以unsigned byte数组形式返回。只读
<code>responseStream</code>	以Ado Stream对象的形式返回响应信息。只读
<code>responseText</code>	将响应信息作为字符串返回。只读
<code>responseXML</code>	将响应信息格式化为Xml Document对象并返回，只读
<code>status</code>	返回当前请求的http状态码。只读
<code>statusText</code>	返回当前请求的响应行状态，只读

\* 表示此属性是W3C文档对象模型的扩展。

## 方法

<a href="#">abort</a>	取消当前请求
<a href="#">getAllResponseHeaders</a>	获取响应的所有http头
<a href="#">getResponseHeader</a>	从响应信息中获取指定的http头
<a href="#">open</a>	创建一个新的http请求，并指定此请求的方法、URL以及验证信息(用户名/密码)
<a href="#">send</a>	发送请求到http服务器并接收回应
<a href="#">setRequestHeader</a>	单独指定请求的某个http头

事件

无

参考

[XMLHttpRequest](#)