

1. XMLHttpRequest的概述

-
- XMLHttpRequest 最早是在IE5中以ActiveX组件的形式实现的。非 W3C 标准.
-
- 创建XMLHttpRequest对象（由于非标准所以实现方法不统一）
 - Internet Explorer把XMLHttpRequest实现为一个ActiveX对象
 - 其他浏览器（Firefox、Safari、Opera...）把它实现为一个本地的JavaScript对象。
 - XMLHttpRequest在不同浏览器上的实现是兼容的，所以可以用同样的方式访问XMLHttpRequest实例的属性和方法，而不论这个实例创建的方法是什么。

2. 创建XMLHttpRequest对象

- 为了每次写Ajax的时候都节省一点时间，可以把对象检测的内容打包成一个可复用的函数：

```
function getHTTPObject() {  
    var xhr = false;  
    if (window.XMLHttpRequest) {  
        xhr = new XMLHttpRequest();  
    } else if (window.ActiveXObject) {  
        xhr = new ActiveXObject("Microsoft.XMLHTTP");  
    }  
    return xhr;  
}
```

说明：对window.XMLHttpRequest的调用会返回一个对象或null，if语句会把调用返回的结果看作是true或false（如果返回对象则为true，返回null则为false）。如果XMLHttpRequest对象存在，则把 xhr 的值设为该对象的新实例。如果不存在，就去检测 ActiveObject 的实例是否存在，如果答案是肯定的，则把微软 XMLHTTP 的新实例赋给 xhr。

3. XMLHttpRequest的方法

方法	描述
abort()	停止当前请求
getAllResponseHeaders()	把HTTP请求的所有响应首部作为键/值对返回
getResponseHeader(“header”)	返回指定首部的串值
open(“method”, “url”)	建立对服务器的调用。Method参数可以是GET、POST或PUT。url参数可以是相对URL或绝对URL。
send(content)	向服务器发送请求
setRequestHeader(“header”, “value”)	把指定首部设置为所提供的值。在设置任何首部之前必须先调用open()

4. XMLHttpRequest的属性

属性	描述
onreadystatechange	每个状态改变是都会触发这个事件处理器，通常会调用一个JavaScript函数
readyState	请求的状态，有5个可取值：0=未初始化、1=正在加载、2=已经加载、3=交互中、4=完成。
responseText	服务器的响应，表示为一个串。
responseXML	服务器的响应，表示为XML。这个对象可以解析为DOM对象。
status	服务器的HTTP状态码（200对应OK、404对应NotFound、等）
statusText	HTTP状态码的相应文本（OK或NotFound等）

5. 发送请求

- 利用XMLHttpRequest 实例与服务器进行通信包含以下3个关键部分：
 - onreadystatechange 事件处理函数
 - open 方法
 - send 方法
- onreadystatechange:
 - 该事件处理函数由服务器触发，而不是用户
 - 在 Ajax 执行过程中，服务器会通知客户端当前的通信状态。这依靠更新XMLHttpRequest 对象的 readyState 来实现。改变 readyState 属性是服务器对客户端连接操作的一种方式。每次 readyState 属性的改变都会触发readystatechange 事件
- open(method, url, asynch)
 - XMLHttpRequest 对象的 open 方法允许程序员用一个Ajax调用向服务器发送请

求。

- method: 请求类型, 类似 “GET”或”POST”的字符串。若只想从服务器检索一个文件, 而不需要发送任何数据, 使用GET(可以在GET请求里通过附加在URL上的查询字符串来发送数据, 不过数据大小限制为2000个字符)。若需要向服务器发送数据, 用POST。
- 在某些情况下, 有些浏览器会把多个XMLHttpRequest请求的结果缓存在同一个URL。如果对每个请求的响应不同, 就会带来不好的结果。在此将时间戳追加到URL的最后, 就能确保URL的唯一性, 从而避免浏览器缓存结果。
- url: 路径字符串, 指向你所请求的服务器上的那个文件。可以是绝对路径或相对路径。
- async: 表示请求是否要异步传输, 默认值为true。指定true, 在读取后面的脚本之前, 不需要等待服务器的相应。指定false, 当脚本处理过程经过这点时, 会停下来, 一直等到Ajax请求执行完毕再继续执行。

- send(data):

- open 方法定义了 Ajax 请求的一些细节。send 方法可为已经待命的请求发送指令
- data: 将要传递给服务器的字符串。
- 若选用的是 GET 请求, 则不会发送任何数据, 给 send 方法传递 null 即可:
request.send(null);
- 当向send()方法提供参数时, 要确保open()中指定的方法是POST, 如果没有数据作为请求体的一部分发送, 则使用null.
- 完整的 Ajax 的 GET 请求示例:

```
var request = getHTTPObject();  
if(request){  
    request.onreadystatechange = doSomething;  
    request.open("GET","file.txt",true);  
    request.send(null);  
}
```

- setRequestHeader(header,value)

- 当浏览器向服务器请求页面时, 它会伴随这个请求发送一组首部信息。这些首部信息是一系列描述请求的元数据(metadata)。首部信息用来声明一个请求是 GET 还是 POST。
- Ajax 请求中, 发送首部信息的工作可以由 setRequestHeader该完成
- 参数header: 首部的名字; 参数value: 首部的值。
- 如果用 POST 请求向服务器发送数据, 需要将 “Content-type” 的首部设置为 “application/x-www-form-urlencoded”. 它会告知服务器正在发送数据, 并且数据已

经符合URL编码了。

- 该方法必须在open()之后才能调用
- 完整的 Ajax 的 POST 请求示例：

```
var url = "../jsp/forumServlet";
var nameValue = trim(document forumAddForm.name.value);
xhr.open("POST",url);
xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
xhr.send("method=name_isExist" + "&name=" + nameValue);
```

6. 接收相应

- 用 XMLHttpRequest 的方法可向服务器发送请求。在 Ajax 处理过程中，XMLHttpRequest 的如下属性可被服务器更改：
 - readyState
 - status
 - responseText
 - responseXML
- - readyState
 - readyState 属性表示Ajax请求的当前状态。它的值用数字代表。
 - 0 代表未初始化。还没有调用 open 方法
 - 1 代表正在加载。open 方法已被调用，但 send 方法还没有被调用
 - 2 代表已加载完毕。send 已被调用。请求已经开始
 - 3 代表交互中。服务器正在发送响应
 - 4 代表完成。响应发送完毕
 - 每次 readyState 值的改变，都会触发 readystatechange 事件。如果把 onreadystatechange 事件处理函数赋给一个函数，那么每次 readyState 值的改变都会引发该函数的执行。
 - readyState 值的变化会因浏览器的不同而有所差异。但是，当请求结束的时候，每个浏览器都会把 readyState 的值统一设为 4
 - status
 - 服务器发送的每一个响应也都带有首部信息。三位数的状态码是服务器发送的响应中最重要的首部信息，并且属于超文本传输协议中的一部分。

- 常用状态码及其含义：
 - 404 没找到页面(not found)
 - 403 禁止访问(forbidden)
 - 500 内部服务器出错(internal service error)
 - 200 一切正常(ok)
 - 304 没有被修改(not modified)
- 在 XMLHttpRequest 对象中，服务器发送的状态码都保存在 status 属性里。通过把这个值和 200 或 304 比较，可以确保服务器是否已发送了一个成功的响应

-

- responseText

- XMLHttpRequest 的 responseText 属性包含了从服务器发送的数据。它是一个 HTML,XML或普通文本，这取决于服务器发送的内容。
- 当 readyState 属性值变成 4 时，responseText 属性才可用，表明 Ajax 请求已经结束。

```
function doSomething(){  
    if(request.readyState == 4){  
        if(request.status == 200 || request.status == 304){  
            alert(request.responseText);  
        }  
    }  
}
```

-

- responseXML

- 如果服务器返回的是 XML，那么数据将储存在 responseXML 属性中。
- 只用服务器发送了带有正确首部信息的数据时，responseXML 属性才是可用的。
MIME 类型必须为 text/xml

