XMLHttpRequest对象的属性和方法

# XMLHttpRequest对象的属性和事件

|  |  |
| --- | --- |
| **属性** | **描述** |
| readyState | 表示XMLHttpRequest对象的状态[1] |
| responseText | 包含客户端接收到的HTTP相应的文本内容[2] |
| responseXML | 服务器响应的XML内容对应的DOM对象[3] |
| status | 服务器返回http状态码[4] |
| statusText | 服务器返回状态码的文本信息[5] |
| **事件** | **描述** |
| onreadystatechange | 当readyState属性发生变化时触发此事件，用于触发回调函数。 |

## XMLHttpRequest对象的状态

|  |  |  |
| --- | --- | --- |
| 状态 | 名称 | 描述 |
| 0 | Uninitialized | 初始化状态。XMLHttpRequest 对象已创建或已被 abort() 方法重置。 |
| 1 | Open | open() 方法已调用，但是 send() 方法未调用。请求还没有被发送。 |
| 2 | Send | Send() 方法已调用，HTTP 请求已发送到 Web 服务器。未接收到响应。 |
| 3 | Receiving | 所有响应头部都已经接收到。响应体开始接收但未完成。 |
| 4 | Loaded | HTTP 响应已经完全接收。 |

## responseText属性

readyState=4时，responseText包含完整的响应信息。

readyState=3时，responseText包含未完整的响应信息。

readyState<3时，responseText为空字符串。

## responseXML

当readyState=4,并且响应头部的Content-Type的MIME类型为XML(text/xml或application/xml)时,该属性有值并且被解析成一个XML文档。其它情况为null,包括回传的XML文档不良或未完成响应回传。

## status

status属性描述了一个HTTP状态代码。仅当服务器readyState属性值为3（正在接受）或4（已加载）时候才能对此属性进行访问。当 readyState 小于 3 的时候读取这一属性会导致一个异常。

## statusText

statusText属性描述了一个HTTP状态代码文本。并且仅当readyState属性值为3或者4时才可用。和 status 属性一样，当 readyState 小于 3 的时候读取这一属性会导致一个异常。

# XMLHttprequest对象的方法

## open方法

描述：制定和服务器交互的HTTP方法，URL地址及其他请求信息。open(method,url, async, username, password) 用来进行初始化工作

返回值：得到一个包含send()方法的对象

参数：有以下五个参数

* method：必须。用于指定HTTP请求方法，支持所有HTTP的方法，如GET,POST，按规定
* uri：请求的服务器的地址，自动解析成绝对地址。
* async：请求是否异步，true表示你异步，false表示同步，默认为true。
* username,password：可以不指定，分别表示用户名和密码，提供HTTP认证机制需要的用户名和密码。

调用open后，readystate状态为1.

## send(content)方法

描述：向服务器发出请求，其内容可以是DOM对象，输入流或是字符串。

调用open 方法后，可以调用send()方法来发送请求。当open 中async=true时，send()方法调用后立即返回，否则会中断直到请求返回。

## abort()方法

该方法可以暂停一个HttpRequest请求或者HttpResponse的接收，并且将XMLHttpRequest的状态设置为初始化。

## setRequestHeader(header,value)方法

该方法用来设置请求的头部信息。当readyState属性值为1时，可以在open方法之后send方法之前调用。

## getResponseHeader()方法

描述：返回包含HTTP的所有响应头信息，其中响应头包括Content-Length，Date，URI等内容。当readystate>2时，该方法用来检索响应的头部信息。否则返回一个空字符串。getAllResponseHeaders()方法返回所有的HttpResponse头部信息。

# Ajax的使用

|  |
| --- |
| function getXhr(){  var xhr = null;  if(window.XMLHttpRequest){  //非ie浏览器  xhr = new XMLHttpRequest();  }else{  //ie浏览器  xhr = new ActiveXObject('MicroSoft.XMLHttp');  }  return xhr;  } |
| step1,获得ajax对象比如:  var xhr = getXhr();  step2,发请求  1)发get请求  //true:表示发送异步请求  //false:表示发送同步请求(发送同步请求时，浏览器会锁定当前页面，用户不能对当前页面做其它操作)。  xhr.open('get', 'check\_username.do?username=tom',true);  //绑订一个事件处理函数  xhr.onreadystatechange=f1;  //send方法会将请求数据包发送出去。  xhr.send(null);  2)发post请求  xhr.open('post', 'check\_username.do',true);  //添加一个content-type消息头  xhr.setRequestHeader('content-type', 'application/x-www-form-urlencoded');  xhr.onreadystatechange=f1;  xhr.send('username=tom');    step3,编写服务器端的处理逻辑，一般只需要返回部分的数据。  step4,编写事件处理函数(处理服务器返回的数据)  function f1(){  //因为只有其值等于4,才能获得服务器  //返回的所有的数据  if(xhr.readyState == 4){  //获得服务器返回的文本  var txt = xhr.responseText;  //更新(dom操作)  ...  }  } |