**Ajax简介**

转载 2016年03月23日 15:19:36

原文：<http://blog.csdn.net/missysm586/article/details/8034241>

http://blog.csdn.net/missysm586/article/details/8090100

一、Ajax概述  
  Ajax的全称是Asynchronous JavaScript and XML,中文定义为”异步JavaScript和XML”，它是Web2.0技术的核心，由多种技术组合而成。使用Ajax技术不必刷新整个页面，只需对页面的局部进行更新，可以节省网络带宽，提高网页加载速度，从而缩短用户等待时间，改善用户体验。  Ajax技术主要包括：客户端脚本语言JavaScript、异步数据获取技术XMLHttpRequest、数据互换和操作技术XML和XSLT、动态显示和交互技术DOM及基于标准的表示技术XHTML和CSS等。Ajax极大的发掘了Web浏览器的潜力，开启了大量的可能性，从而有效地改善了用户操作体验。     
  传统的web应用允许用户填写表单(form)，当提交表单时就向web服务器发送一个请求。服务器接收并处理传来的表单，然後返回一个新的网页。这个做法 浪费了许多带宽，因为在前後两个页面中的大部分HTML代码往往是相同的。由于每次应用的交互都需要向服务器发送请求，应用的响应时间就依赖于服务器的响 应时间。这导致了用户界面的响应比本地应用慢得多。  
  与此不同，AJAX应用可以仅向服务器发送并取回必需的数据， 它使用SOAP或其它一些基于XML的web service接口，并在客户端采用JavaScript处理来自服务器的响应。因为在服务器和浏览器之间交换的数据大量减少，结果我们就能看到响应更快的应用。同时很多的处理工作可以在发出请求的客户端机器上完成，所以Web服务器的处理时间也减少了。  
   
**二、Ajax的工作原理**  
  
　　Ajax的核心是JavaScript对象XmlHttpRequest。该对象在Internet Explorer 5中首次引入，它是一种支持异步请求的技术。简而言之，XmlHttpRequest使您可以使用JavaScript向服务器提出请求并处理响应，而不阻塞用户。  
  
 在创建Web站点时，在客户端执行屏幕更新为用户提供了很大的灵活性。下面是使用Ajax可以完成的功能：  
  
动态更新购物车的物品总数，无需用户单击Update并等待服务器重新发送整个页面。  
提升站点的性能，这是通过减少从服务器下载的数据量而实现的。例如，在Amazon的购物车页面，当更新篮子中的一项物品的数量时，会重新载入整个页面，这必须下载 32K的数据。如果使用Ajax计算新的总量，服务器只会返回新的总量值，因此所需的带宽仅为原来的百分之一。  
消除了每次用户输入时的页面刷新。例如，在Ajax中，如果用户在分页列表上单击Next，则服务器数据只刷新列表而不是整个页面。  
直接编辑表格数据，而不是要求用户导航到新的页面来编辑数据。对于Ajax，当用户单击Edit时，可以将静态表格刷新为内容可编辑的表格。用户单击Done之后，就可以发出一个Ajax请求来更新服务器，并刷新表格，使其包含静态、只读的数据。

三、简单使用

Ajax的核心是JavaScript对象XmlHttpRequest。

首先，您拥有一些简单的定位点标记，它连接到一个JavaScriptcastVote(rank)函数。

**[javascript]** [view plain](http://blog.csdn.net/hudashi/article/details/50963494) [copy](http://blog.csdn.net/hudashi/article/details/50963494)

1. **function** castVote(rank) {
2. **var** url = "/ajax-demo/static-article-ranking.html";
3. **var** callback = processAjaxResponse;
4. executeXhr(callback, url);
5. }

　　该函数为您想要与之通信的服务器资源创建一个URL并调用内部函数executeXhr，提供一个回调JavaScript函数，一旦服务器响应可用，该函数就被执行。由于我希望它运行在一个简单的Apache环境中，“cast vote URL”只是一个简单的HTML页面。在实际情况中，被调用的URL将记录票数并动态地呈现包含投票总数的响应。

　　下一步是发出一个XmlHttpRequest请求：

**[javascript]** [view plain](http://blog.csdn.net/hudashi/article/details/50963494) [copy](http://blog.csdn.net/hudashi/article/details/50963494)

1. **function** executeXhr(callback, url) {
2. // branch for native XMLHttpRequest object
3. **if** (window.XMLHttpRequest) {
4. req = **new** XMLHttpRequest();
5. req.onreadystatechange = callback;
6. req.open("GET", url, **true**);
7. req.send(**null**);
8. } // branch for IE/Windows ActiveX version
9. **else** **if** (window.ActiveXObject) {
10. req = **new** ActiveXObject("Microsoft.XMLHTTP");
11. **if** (req) {
12. req.onreadystatechange = callback;
13. req.open("GET", url, **true**);
14. req.send();
15. }
16. }
17. }

　　如您所见，执行一个XmlHttpRequest并不简单，但非常直观。和平常一样，在JavaScript领域，大部分的工作量都花在确保浏览器兼容方面。在这种情况下，首先要确定XmlHttpRequest是否可用。如果不能用，很可能要使用Internet Explorer，这样就要使用所提供的ActiveX实现。

executeXhr()方法中最关键的部分是这两行：

req.onreadystatechange = callback;  
req.open("GET", url, true);

　　第一行定义了JavaScript回调函数，您希望一旦响应就绪它就自动执行，而req.open()方法中所指定的“true”标志说明您想要异步执行该请求。

　　一旦服务器处理完XmlHttpRequest并返回给浏览器，使用req.onreadystatechange指派所设置的回调方法将被自动调用。

**[javascript]** [view plain](http://blog.csdn.net/hudashi/article/details/50963494) [copy](http://blog.csdn.net/hudashi/article/details/50963494)

1. **function** processAjaxResponse() {
2. // only if req shows "loaded"
3. **if** (req.readyState == 4) {
4. // only if "OK"
5. **if** (req.status == 200) {
6. $('votes').innerHTML = req.responseText;
7. } **else** {
8. alert("There was a problem retrieving the XML data:\n" +
9. req.statusText);
10. }
11. }
12. }

　　该代码相当简洁，并且使用了几个函数，这使得难以一下子看出发生了什么。为了弄清楚这一点，下面的表格，列举了常用的XmlHttpRequest对象属性。

|  |  |
| --- | --- |
| **属性** | **描述** |
| onreadystatechange | 每次状态改变所触发事件的事件处理程序 |
| readyState | 对象状态值：   * 0 = 未初始化（uninitialized） * 1 = 正在加载（loading） * 2 = 加载完毕（loaded） * 3 = 交互（interactive） * 4 = 完成（complete） |
| responseText | 从服务器进程返回的数据的字符串形式 |
| responseXML | 从服务器进程返回的DOM兼容的文档数据对象 |
| status | 从服务器返回的数字代码，比如404（未找到）或200（就绪） |
| statusText | 伴随状态码的字符串信息 |

　　现在processVoteResponse()函数开始显示出其意义了。它首先检查XmlHttpRequest的整体状态以保证它已经完成（readyStatus == 4），然后根据服务器的设定询问请求状态。如果一切正常（status == 200）,就使用innerHTML属性重写DOM的“votes”节点的内容。