昵称：飞鱼在天
园龄：9个月
粉丝：0
关注：1
+加关注

< 2018年3月 >

日	一	二	三	四	五	六
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

随笔档案

2017年7月 (2)

2017年6月 (6)

最新评论

1. Re:ArrayList概述

我问问

--whp1992

阅读排行榜

1. Ajax工作原理及实例(1480)

2. mybatis和ibatis区别(23)

3. 乐观锁和悲观锁(16)

Ajax工作原理及实例

1、关于ajax的名字

ajax 的全称是Asynchronous JavaScript and XML，其中，Asynchronous 是异步的意思，它有别于传统web开发中采用的同步的方式。

2、关于同步和异步

步传输是面向字符的传输，它的单位是字符；而同步传输是面向比特的传输，它的单位是桢，它传输的时候要求接受方和发送方的时钟是保持一致的。

具体来说，异步传输是将比特分成小组来进行传送。一般每个小组是一个8位字符，在每个小组的头部和尾部都有一个开始位和一个停止位，它在传送过程中接收方和发送方的时钟不要求一致，也就是说，发送方可以在任何时刻发送这些小组，而接收方并不知道它什么时候到达。一个最明显的例子就是计算机键盘和主机的通信，按下一个键的同时向主机发送一个8比特位的ASCII代 码，键盘可以在任何时刻发送代码，这取决于用户的输入速度，内部的硬件必须能够在任何时刻接收一个键入的字符。这是一个典型的异步传输过程。异步传输存在 一个潜在的问题，即接收方并不知道数据会在什么时候到达。在它检测到数据并做出响应之前，第一个比特已经过去了。这就像有人出乎意料地从后面走上来跟你说话，而你没来得及反应过来，漏掉了最前面的几个词。因此，每次异步传输的信息都以一个起始位开头，它通知接收方数据已经到达了，这就给了接收方响应、接收 和缓存数据比特的时间；在传输结束时，一个停止位表示该次传输信息的终止。按照惯例，空闲（没有传送数据）的线路实际携带着一个代表二进制1的信号。步传输的开始位使信号变成0，其他的比特位使信号随传输的数据信息而变化。最后，停止位使信号重新变回1，该信号一直保持到下一个开始位到达。例如在键盘上数字“1”，按照8比特位的扩展ASCII编码，将发送“00110001”，同时需要在8比特位的前面加一个起始位，后面一个停止位。

同步传输的比特分组要大得多。它不是独立地发送每个字符，每个字符都有自己的开始位和停止位，而是把它们组合起来一起发送。我们将这些组合称为数据帧，或简称为帧。

数据帧的第一部分包含一组同步字符，它是一个独特的比特组合，类似于前面提到的起始位，用于通知接收方一个帧已经到达，但它同时还能确保接收方的采样速度和比特的到达速度保持一致，使收发双方进入同步。

帧的最后一部分是一个帧结束标记。与同步字符一样，它也是一个独特的比特串，类似于前面提到的停止位，用于表示在下一帧开始之前没有别的即将到达的数据了。

同步传输通常要比异步传输快速得多。接收方不必对每个字符进行开始和停止的操作。一旦检测到帧同步字符，它就在接下来的数据到达时接收它们。另外，同步传输的开销也比较少。例如，一个典型的帧可能有500字节（即4000比特）的数据，其中可能只包含100比特的开销。这时，增加的比特位使传输的比特总数增加2.5%，这与异步传输中25 %的增值要小得多。随着数据帧中实际数据比特位的增加，开销比特所占的百分比将相应地减少。但是，数据比特位越长，缓存数据所需要的缓冲区也越大，这就限制了一个帧的大小。另外，帧越大，它占据传输媒体的连续时间也越长。在极端的情况下，这将导致其他用户等得太久。

3、**ajax所包含的技术**

ajax并非一种新的技术，而是几种原有技术的结合体。它由下列技术组合而成。

1.使用CSS和XHTML来表示。

2. 使用DOM模型来交互和动态显示。

3.使用XMLHttpRequest来和服务器进行异步通信。

4.使用javascript来绑定和调用。

在上面几中技术中，除了XmlHttpRequest对象以外，其它所有的技术都是基于web标准并且已经得到了广泛使用的，XMLHttpRequest虽然目前还没有被W3C所采纳，但是它已经是一个事实的标准，因为目前几乎所有的主流浏览器都支持它。

4、ajax原理和XmlHttpRequest对象

Ajax的原理简单来说通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用javascript来操作DOM而更新页面。这其中最关键的一步就是从服务器获得请求数据。要清楚这个过程和原理，我们必须对 XMLHttpRequest有所了解。

4. ArrayList概述(16)

5. CAS实现单点登录理解(14)

评论排行榜

1. ArrayList概述(1)

XMLHttpRequest是ajax的核心机制，它是在IE5中首先引入的，是一种支持异步请求的技术。简单的说，也就是javascript可以及时向服务器提出请求和处理响应，而不阻塞用户。达到无刷新的效果。

所以我们先从XMLHttpRequest讲起，来看看它的工作原理。

首先，我们先来看看XMLHttpRequest这个对象的属性。

它的属性有：

onreadystatechange 每次状态改变所触发事件的事件处理程序。

responseText 从服务器进程返回数据的字符串形式。

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，比如常见的404（未找到）和200（已就绪）

status Text 伴随状态码的字符串信息

readyState 对象状态值

0 (未初始化) 对象已建立，但是尚未初始化（尚未调用open方法）

1 (初始化) 对象已建立，尚未调用send方法

2 (发送数据) send方法已调用，但是当前的状态及http头未知

3 (数据传送中) 已接收部分数据，因为响应及http头不全，这时通过responseBody和responseText获取部分数据会出现错误，

4 (完成) 数据接收完毕,此时可以通过通过responseXml和responseText获取完整的回应数据

但是，由于各浏览器之间存在差异，所以创建一个XMLHttpRequest对象可能需要不同的方法。这个差异主要体现在IE和其它浏览器之间。下面是一个比较标准的创建XMLHttpRequest对象的方法。

```
function CreateXmlHttp() {

    //非IE浏览器创建XmlHttpRequest对象
    if (window.XmlHttpRequest) {
        xmlhttp = new XmlHttpRequest();
    }

    //IE浏览器创建XmlHttpRequest对象
    if (window.ActiveXObject) {
        try {
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch (e) {
            try {
                xmlhttp = new ActiveXObject("msxml2.XMLHTTP");
            }
            catch (ex) { }
        }
    }
}

function Ustbwuyi() {

    var data = document.getElementById("username").value;
    CreateXmlHttp();
    if (!xmlhttp) {
        alert("创建xmlhttp对象异常！");
        return false;
    }

    xmlhttp.open("POST", url, false);

    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState == 4) {
            document.getElementById("user1").innerHTML = "数据正在加载...";
            if (xmlhttp.status == 200) {
                document.write(xmlhttp.responseText);
```

```
            }
        }
    }
    xmlhttp.send();
}
```

如上所示，函数首先检查XMLHttpRequest的整体状态并且保证它已经完成（readyStatus=4），即数据已经发送完毕。然后根据服务器的设定询问请求状态，如果一切已经就绪（status=200），那么就执行下面需要的操作。

对于XmlHttpRequest的两个方法，open和send，其中open方法指定了：

a、向服务器提交数据的类型，即post还是get。

b、请求的url地址和传递的参数。

c、传输方式，false为同步，true为异步。默认为true。如果是异步通信方式(true)，客户机就不等待服务器的响应；如果是同步方式(false)，客户机就要等到服务器返回消息后才去执行其他操作。我们需要根据实际需要来指定同步方式，在某些页面中，可能会发出多个请求，甚至是有组织有计划有队形大规模的高强度的request，而后一个是会覆盖前一个的，这个时候当然要指定同步方式。

Send方法用来发送请求。

知道了XMLHttpRequest的工作流程，我们可以看出，XMLHttpRequest是完全用来向服务器发出一个请求的，它的作用也局限于此，但它的作用是整个ajax实现的关键，因为ajax无非是两个过程，发出请求和响应请求。并且它完全是一种客户端的技术。而XMLHttpRequest正是处理了服务器端和客户端通信的问题所以才会如此的重要。

现在，我们对ajax的原理大概可以有一个了解了。我们可以把服务器端看成一个数据接口，它返回的是一个纯文本流，当然，这个文本流可以是XML格式，可以是Html，可以是Javascript代码，也可以只是一个字符串。这时候，XMLHttpRequest向服务器端请求这个页面，服务器端将文本的结果写入页面，这和普通的web开发流程是一样的，不同的是，客户端在异步获取这个结果后，不是直接显示在页面，而是先由javascript来处理，然后再显示在页面。至于现在流行的很多ajax控件，比如magicajax等，可以返回DataSet等其它数据类型，只是将这个过程封装了的结果，本质上他们并没有什么太大的区别。

5、ajax的缺点

下面我着重讲一讲ajax的缺陷，因为平时我们大多注意的都是ajax给我们所带来的好处诸如用户体验的提升。而对ajax所带来的缺陷有所忽视。

下面所阐述的ajax的缺陷都是它先天所产生的。

1、ajax干掉了back按钮，即对浏览器后退机制的破坏。后退按钮是一个标准的web站点的重要功能，但是它没法和js进行很好的合作。这是ajax所带来的一个比较严重的问题，因为用户往往是希望能够通过后退来取消前一次操作的。那么对于这个问题有没有办法？答案是肯定的，用过Gmail的知道，Gmail下面采用的ajax技术解决了这个问题，在Gmail下面是可以后退的，但是，它也并不能改变ajax的机制，它只是采用的一个比较笨但是有效的办法，即用户单击后退按钮访问历史记录时，通过创建或使用一个隐藏的IFRAME来重现页面上的变更。（例如，当用户在Google Maps中单击后退时，它在一个隐藏的IFRAME中进行搜索，然后将搜索结果反映到Ajax元素上，以便将应用程序状态恢复到当时的状态。）

但是，虽然说这个问题是可以解决的，但是它所带来的开发成本是非常高的，和ajax框架所要求的快速开发是相背离的。这是ajax所带来的一个非常严重的问题。

2、安全问题

技术同时也对IT企业带来了新的安全威胁，ajax技术就如同对企业数据建立了一个直接通道。这使得开发者在不经意间会暴露比以前更多的数据和服务器逻辑。ajax的逻辑可以对客户端的安全扫描技术隐藏起来，允许黑客从远端服务器上建立新的攻击。还有ajax也难以避免一些已知的安全弱点，诸如跨站点脚步攻击、SQL注入攻击和基于credentials的安全漏洞等。

3、对搜索引擎的支持比较弱。

4、破坏了程序的异常机制。至少从目前看来，像ajax.dll，ajaxpro.dll这些ajax框架是会破坏程序的异常机制的。关于这个问题，我曾经在开发过程中遇到过，但是查了一下网上几乎没有相关的介绍。后来我自己做了一次试验，分别采用ajax和传统的form提交的模式来删除一条数据……给我们的调试带来了很大的困难。

5、另外，像其他方面的一些问题，比如说违背了url和资源定位的初衷。例如，我给你一个url地址，如果采用了ajax技术，也许你在该url地址下面看到的和我在这个url地址下看到的内容是不同的。这个和资源定位的初衷是相背离的。

6、一些手持设备（如手机、PDA等）现在还不能很好的支持ajax，比如说我们在手机的浏览器上打开采用ajax技术的网站时，它目前是不支持的，当然，这个问题和我们没太多关系。

5、$.ajax()方法详解

jquery中的ajax方法参数总是记不住，这里记录一下。

1.url:
要求为String类型的参数，（默认为当前页地址）发送请求的地址。

2.type:
要求为String类型的参数，请求方式（post或get）默认为get。注意其他http请求方法，例如put和delete也可以使用，但仅部分浏览器支持。

3.timeout:
要求为Number类型的参数，设置请求超时时间（毫秒）。此设置将覆盖$.ajaxSetup()方法的全局设置。

4.async:
要求为Boolean类型的参数，默认设置为true，所有请求均为异步请求。如果需要发送同步请求，请将此选项设置为false。注意，同步请求将锁住浏览器，用户其他操作必须等待请求完成才可以执行。

5.cache:
要求为Boolean类型的参数，默认为true（当dataType为script时，默认为false），设置为false将不会从浏览器缓存中加载请求信息。

6.data:
要求为Object或String类型的参数，发送到服务器的数据。如果已经不是字符串，将自动转换为字符串格式。get请求中将附加在url后。防止这种自动转换，可以查看 processData选项。对象必须为key/value格式，例如{foo1:"bar1",foo2:"bar2"}转换为&foo1=bar1&foo2=bar2。如果是数组，JQuery将自动为不同值对应同一个名称。例如{foo:["bar1","bar2"]}转换为&foo=bar1&foo=bar2。

7.dataType:
要求为String类型的参数，预期服务器返回的数据类型。如果不指定，JQuery将自动根据http包mime信息返回responseXML或responseText，并作为回调函数参数传递。可用的类型如下：
xml：返回XML文档，可用JQuery处理。
html：返回纯文本HTML信息；包含的script标签会在插入DOM时执行。
script：返回纯文本JavaScript代码。不会自动缓存结果。除非设置了cache参数。注意在远程请求时（不在同一个域下），所有post请求都将转为get请求。
json：返回JSON数据。
jsonp：JSONP格式。使用SONP形式调用函数时，例如myurl?callback=?，JQuery将自动替换后一个“?”为正确的函数名，以执行回调函数。
text：返回纯文本字符串。

8.beforeSend：
要求为Function类型的参数，发送请求前可以修改XMLHttpRequest对象的函数，例如添加自定义HTTP头。在beforeSend中如果返回false可以取消本次ajax请求。XMLHttpRequest对象是惟一的参数。

```
function(XMLHttpRequest){
  this;  //调用本次ajax请求时传递的options参数
}
```

9.complete：
要求为Function类型的参数，请求完成后调用的回调函数（请求成功或失败时均调用）。参数：XMLHttpRequest对象和一个描述成功请求类型的字符串。

```
function(XMLHttpRequest, textStatus){
  this;   //调用本次ajax请求时传递的options参数
}
```

10.success：要求为Function类型的参数，请求成功后调用的回调函数，有两个参数。
(1)由服务器返回，并根据dataType参数进行处理后的数据。
(2)描述状态的字符串。

```
function(data, textStatus){
  //data可能是xmlDoc、jsonObj、html、text等等
  this; //调用本次ajax请求时传递的options参数
}
```

11.error:
要求为Function类型的参数，请求失败时被调用的函数。该函数有3个参数，即XMLHttpRequest对象、错误信息、捕获的错误对象(可选)。ajax事件函数如下：

```
function(XMLHttpRequest, textStatus, errorThrown){
  //通常情况下textStatus和errorThrown只有其中一个包含信息
```

```
        this;  //调用本次ajax请求时传递的options参数
    }
```

12.contentType：
要求为String类型的参数，当发送信息至服务器时，内容编码类型默认为"application/x-www-form-urlencoded"。该默认值适合大多数应用场合。

13.dataFilter：
要求为Function类型的参数，给Ajax返回的原始数据进行预处理的函数。提供data和type两个参数。data是Ajax返回的原始数据，type是调用jQuery.ajax时提供的dataType参数。函数返回的值将由jQuery进一步处理。

```
        function(data, type){
            //返回处理后的数据
            return data;
        }
```

14.dataFilter：
要求为Function类型的参数，给Ajax返回的原始数据进行预处理的函数。提供data和type两个参数。data是Ajax返回的原始数据，type是调用jQuery.ajax时提供的dataType参数。函数返回的值将由jQuery进一步处理。

```
        function(data, type){
            //返回处理后的数据
            return data;
        }
```

15.global：
要求为Boolean类型的参数，默认为true。表示是否触发全局ajax事件。设置为false将不会触发全局ajax事件，ajaxStart或ajaxStop可用于控制各种ajax事件。

16.ifModified：
要求为Boolean类型的参数，默认为false。仅在服务器数据改变时获取新数据。服务器数据改变判断的依据是Last-Modified头信息。默认值是false，即忽略头信息。

17.jsonp：
要求为String类型的参数，在一个jsonp请求中重写回调函数的名字。该值用来替代在"callback=?"这种GET或POST请求中URL参数里的"callback"部分，例如{jsonp:'onJsonPLoad'}会导致将"onJsonPLoad=?"传给服务器。

18.username：
要求为String类型的参数，用于响应HTTP访问认证请求的用户名。

19.password：
要求为String类型的参数，用于响应HTTP访问认证请求的密码。

20.processData：
要求为Boolean类型的参数，默认为true。默认情况下，发送的数据将被转换为对象（从技术角度来讲并非字符串）以配合默认内容类型"application/x-www-form-urlencoded"。如果要发送DOM树信息或者其他不希望转换的信息，请设置为false。

21.scriptCharset：
要求为String类型的参数，只有当请求时dataType为"jsonp"或者"script"，并且type是GET时才会用于强制修改字符集(charset)。通常在本地和远程的内容编码不同时使用。

案例代码：

```
$(function(){
    $('#send').click(function(){
         $.ajax({
             type: "GET",
             url: "test.json",
             data: {username:$("#username").val(), content:$("#content").val()},
             dataType: "json",
             success: function(data){
                        $('#resText').empty();   //清空resText里面的所有内容
                        var html = '';
                        $.each(data, function(commentIndex, comment){
                              html += '<div class="comment"><h6>' +
comment['username']
                                          + ':</h6><p class="para"' +
comment['content']
                                          + '</p></div>';
                        });
```

```
                        $('#resText').html(html);
                    }
            });
    });
});
```

—





飞鱼在天
关注 - 1
粉丝 - 0
+加关注

0 0

« 上一篇：乐观锁和悲观锁
» 下一篇：CAS实现单点登录理解

posted @ 2017-06-12 10:11 飞鱼在天 阅读(1483) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

最新IT新闻:
· 11亿美元重注斗鱼和虎牙后，腾讯已掐住游戏行业的命门
· Twitter将"准直播"世界杯 另买下美足联版权
· 亚马逊对印度寄予厚望 视频服务将增加更多印度内容
· 这家公司向犯罪分子提供定制黑莓手机 CEO被FBI逮捕
· 谷歌和苹果竞相收购最有发展前途的AI初创公司
» 更多新闻...

最新知识库文章:
· 写给自学者的入门指南
· 和程序员谈恋爱
· 学会学习
· 优秀技术人的管理陷阱
· 作为一个程序员，数学对你到底有多重要
» 更多知识库文章...