# 什么是Ajax

**杰西·詹姆斯·加勒特**

Asynchronous JavaScript and XML（异步JavaScript和XML)

这一技术能够向服务器请求额外的数据而无须卸载（刷新）页面，会带来更好的用户体验。

# Ajax的核心

Ajax 技术的核心是XMLHttpRequest 对象（简称XHR），这是由微软首先引入的一个特性，其他浏览器提供商后来都提供了相同的实现

虽然名字中包含XML 的成分，但Ajax 通信与数据格式无关；这种技术就是无须刷新页面即可从服务器取得数据，但不一定是XML 数据。

# Ajax的实现原理

1. 用XHR 向服务器发送请求
2. 使用XHR 对象取得新数据
3. 然后再通过DOM 将新数据插入到页面中

**熟练使用XHR 对象已经成为所有Web 开发人员必须掌握的一种技能。**

# XMLHttpRequest 对象

IE7+、Firefox、Opera、Chrome 和Safari 都支持原生的XHR 对象，在这些浏览器中创建XHR 对象，要像下面这样使用XMLHttpRequest 构造函数。

var xhr = new XMLHttpRequest();

***\*\*\*\*如何兼容IE6创建xhr对象\*\*\*\****

**var xhr=new ActiveXObject(“Microsfot.XMLHttp”);**

# XHR对象的方法

1用xhr.open(method, url,boolean);方法启动一个请求；

2用xhr.send(null);方法 发送一个请求；

两个方法的参数介绍：method=get/post, url不能跨域，Boolean=true/false异步/同步null是为了跨浏览器兼容

# XHR 对象的属性

1、responseText：作为响应主体被返回的文本。

2、responseXML：如果响应的内容类型是"text/xml"或"application/xml"，这个属性中将保存包含着响应数据的XML DOM 文档。

3、status：响应的HTTP 状态。

4、statusText：HTTP 状态的说明。

5、readyState : 该属性表示请求/响应过程的当前活动阶段

 0：未初始化。尚未调用open()方法。

 1：启动。已经调用open()方法，但尚未调用send()方法。

 2：发送。已经调用send()方法，但尚未接收到响应。

 3：接收。已经接收到部分响应数据。

 4：完成。已经接收到全部响应数据，而且已经可以在客户端使用了

6、onreadystatechange 事件属性，以利用这个事件来检测每次状态变化后readyState 的值

**必须在调用open()之前指定onreadystatechange事件处理程序才能确保跨浏览器兼容性**

xhr.onreadystatechange = function(){

if (xhr.readyState == 4){

if (xhr.status == 200){

alert(xhr.responseText);

} else {

alert("请求失败: " + xhr.status);

}

}

};

xhr.open("get", "example.txt", true);

xhr.send(null);

**这个例子在onreadystatechange 事件处理程序中使用了xhr 对象，没有使用**

**this 对象，原因是onreadystatechange 事件处理程序的作用域问题。如果使用**

**this 对象，在有的浏览器中会导致函数执行失败，或者导致错误发生。因此，使用**

**实际的XHR 对象实例变量是较为可靠的一种方式。**

# 获取服务器端的文本文件中的数据

**a.txt 文件中的数据**

abcd

**b.txt文件中存放的数据**

["100","200","300"]

**c.txt文件中存放的数据**

{"name":"lc","sex":"man","age":30}

**d.txt文件中存放的数据**

{”tel”:[”苹果”, ”OPPO”, ”小米”, ”华为”],”price”:[8000,3000,3000,4000]}

***\*\*\*这里存放数据，除了数字和null不用双引号，其他数据都得用双引号\*\*\****

# 封装ajax函数

function ajax(method,url,callBack)

{

var xhr=new XMLHttpRequest();

xhr.onreadystatechange=function()

{

if(xhr.readyState==4)

{

if(xhr.status==200)

{

callBack(xhr.responseText);

}else{

alert('error');

}

}

}

xhr.open(method,url);

xhr.send(null);

}

# 案例模仿百度输入关键字

ajax('get','a.txt',function(data){

var oText=document.getElementById('text');

var oDiv=document.getElementById('div');

var arr=JSON.parse(data);

var str="";

oText.oninput=function()

{

str="";

for(var i=0; i<arr.length; i++)

{

if(arr[i].charAt(0)==oText.value.charAt(0))

{

str+=arr[i]+"<br>"

}

}

oDiv.innerHTML=str;

}

oText.onblur=function()

{

oDiv.innerHTML="";

}

});

# 案例模仿检测账号是否被占用

<input type="text" id="text"><span id="span"></span>

<script>

var oText=document.getElementById('text');

var oSpan=document.getElementById('span');

ajax('get','user.txt',function(data){

/\* oText.oninput=function()

{

oSpan.innerHTML="";

if(data.indexOf(oText.value)!=-1)

{

oSpan.innerHTML="此用户名被占用";

}

} \*/

var arr=data.split("|");

//console.log(arr);

oText.onkeyup=function()

{

oSpan.innerHTML="";

for(var i=0; i<arr.length; i++)

{

if(oText.value==arr[i])

{

oSpan.innerHTML="此用户名被占用";

}

}

}

});

</script>

# 获取服务器端的html文件中的数据

# JSONP(跨域)

JSON with Padding

1.script标签

2.用script标签加载资源是没有跨域问题的

<script>

function fn(data) {

    alert(data);

}

</script>

<script src="2.txt"></script>

1、在资源加载进来之前定义好一个函数，这个函数接收一个参数（数据），函数里面利用这个参数做一些事情

2、然后需要的时候通过script标签加载对应远程文件资源，当远程的文件资源被加载进来的时候，就会去执行我们前面定义好的函数，并且把数据当作这个函数的参数传入进去

**当按钮点击的时候再去加载远程资源，让他执行**

<script>

function fn(data) {

    alert(data);

}

</script>

<script>

window.onload = function() {

    var oBtn = document.getElementById('btn');

    oBtn.onclick = function() {

        var oScript = document.createElement('script');

        oScript.src = '2.txt';

        document.body.appendChild(oScript);

    }

}

</script>

# 案例真实百度输入关键字搜索

<script>

function callback(data) {

    var oUl = document.getElementById('ul1');

    var html = '';

    if (data.s.length) {

        oUl.style.display = 'block';

        for (var i=0; i<data.s.length; i++) {

            html += '<li><a target="\_blank" href="http://www.baidu.com/s?wd='+data.s[i]+'">'+ data.s[i] +'</a></li>';

        }

        oUl.innerHTML = html;

    } else {

        oUl.style.display = 'none';

    }

}

window.onload = function() {

    var oQ = document.getElementById('q');

    var oUl = document.getElementById('ul1');

    oQ.onkeyup = function() {

        if ( this.value != '' ) {

            var oScript = document.createElement('script');

            oScript.src = 'http://suggestion.baidu.com/su?wd='+this.value+'&cb=callback';

            document.body.appendChild(oScript);

        } else {

            oUl.style.display = 'none';

        }

    }

}

</script>

# 案例豆瓣图书搜索

<script>

function fn1(data) {

    var oMsg = document.getElementById('msg');

    var oList = document.getElementById('list');

    console.log(data);

    oMsg.innerHTML = data.title.$t + ' : ' + data['opensearch:totalResults'].$t;

    var aEntry = data.entry;

    var html = '';

    for (var i=0; i<aEntry.length; i++) {

        html += '<dl><dt>'+ aEntry[i].title.$t +'</dt><dd><img src="'+ aEntry[i].link[2]['@href'] +'" /></dd></dl>';

    }

    oList.innerHTML = html;

}

window.onload = function() {

    var oQ = document.getElementById('q');

    var oBtn = document.getElementById('btn');

    var oMsg = document.getElementById('msg');

    var oList = document.getElementById('list');

    oBtn.onclick = function() {

        if ( oQ.value != '' ) {

            var oScript = document.createElement('script');

            oScript.src = 'http://api.douban.com/book/subjects?q='+oQ.value+'&alt=xd&callback=fn1';

            document.body.appendChild(oScript);

        }

    }

}

</script>

</head>

<body>

    豆瓣图书搜索

    <input type="text" id="q" /><input type="button" id="btn" value="搜索" />

<p id="msg"></p>

<hr />

    <div id="list"></div>

</body>

# 获取服务器端的php文件中的数据

1. php语言的简单介绍
   1. echo
   2. 字符串拼接 .号 {}号
   3. header('content-type:text/html;charset="utf-8"')
   4. array()
   5. var\_dump()
   6. print\_r()
   7. json\_encode()
2. 字符串数据
3. 数组数据
4. json数据
5. get带数据请求 $\_GET[] isset()
6. post带数据请求 $\_POST[]
7. xhr对象的setRequestHeader('content-type','application/x-www-form-urlencoded')
8. get
   1. 1缓存会出问题new Date().getTime()
   2. 2中文要进行编码 encodeURI (有些浏览器地址栏只支持ASCII码)
9. 获取新闻列表案例

**get带数据请求**

var oText=document.getElementById('text');

var oBtn=document.getElementById('btn');

    oBtn.onclick=function ()

    {

var oAjax=new XMLHttpRequest();

        oAjax.open('GET','5demo-ajax-get.php?username='+encodeURI(oText.value)+'&age=20&'+new Date().getTime(),true);

oAjax.send();

        oAjax.onreadystatechange=function()

        {

            if(oAjax.readyState==4)

            {

                if(oAjax.status==200)

{

alert(oAjax.responseText);

}else

{

alert('出错了：Err'+oAjax.status);

}

            }

        }

    }

<?php

header('content-type:text/html;charset="utf-8"');

error\_reporting(0);

$username=$\_GET['username'];

$age=$\_GET['age'];

echo "您的名字是：{$username},您的年龄是：{$age}";

?>

**post带数据请求 $\_POST[]**

var oBtn=document.getElementById('btn');

    oBtn.onclick=function ()

    {

var oAjax=new XMLHttpRequest();

        oAjax.open('post','ajax-post.php',true);

//告诉服务器，通过请求头，URL编码后发送给服务器

oAjax.setRequestHeader('content-type','application/x-www-form-urlencoded');

//post方式，数据放在send方法里作为参数传递

oAjax.send('username=传习&age=20');

        oAjax.onreadystatechange=function()

        {

            if(oAjax.readyState==4)

            {

                if(oAjax.status==200)

{

alert(oAjax.responseText);

}else

{

alert('出错了：Err'+oAjax.status);

}

            }

        }

}

<?php

header('content-type:text/html;charset="utf-8"');

error\_reporting(0);

$username=$\_POST['username'];

$age=$\_POST['age'];

echo "欢迎，您的名字是：{$username},您的年龄是：{$age}";

?>

**获取新闻列表案例**

    var oBtn=document.getElementById('btn');

    oBtn.onclick=function ()

    {

ajax();

    }

    setInterval(ajax,10000);

    function ajax()

    {

        var oAjax=new XMLHttpRequest();

        oAjax.open('GET','8demo-getNews.php',true);

        oAjax.send();

        oAjax.onreadystatechange=function()

        {

            if(oAjax.readyState==4)

            {

                if(oAjax.status==200)

                {

                    var sData=JSON.parse(oAjax.responseText);

                    var oUl=document.getElementById('ul1');

                    var sHtml='';

                    for(var i=0; i<sData.length; i++)

                    {

                        sHtml+='<li><a href="javascript:;">'+sData[i].title+'</a>[<span>'+sData[i].date+'</span>]</li>';

                    }

                    oUl.innerHTML=sHtml;

                }else

                {

                    alert('出错了：Err'+oAjax.status);

                }

            }

        }

    }

<?php

header('content-type:text/html;charset="utf-8"');

$news=array(

array('title'=>'火箭军进行夜间发射训练 场面似科幻大片(图)','date'=>'2015-1-6'),

array('title'=>'谷歌或将回归中国:发布上海北京60个工作岗位','date'=>'2015-1-6'),

array('title'=>'A股暴跌7.21%再次引发熔断 今日提前收盘','date'=>'2015-1-6'),

array('title'=>'火箭军进行夜间发射训练 场面似科幻大片(图)','date'=>'2015-1-6')

);

echo json\_encode($news);

?>

# 再次封装ajax

function ajax(method,url,data,fnSucc,fnFailed)

{

    if(window.XMLHttpRequest)

    {

        var oAjax=new XMLHttpRequest();

    }else

    {

        var oAjax=new ActiveXObject('Microsoft.XMLHttp');

    }

if(method=='get'&&data)

{

url+='?'+data;

}

    oAjax.open(method,url,true);

if(method=='get')

{

oAjax.send();

}else

{

oAjax.setRequestHeader('content-type','application/x-www-form-urlencoded');

oAjax.send(data);

}

    oAjax.onreadystatechange=function()

    {

        if(oAjax.readyState==4)

        {

            if(oAjax.status==200)

            {

                fnSucc(oAjax.responseText);

            }else

            {

                if(fnFailed)

                {

                    fnFailed(oAjax.status);

                }

            }

        }

    }

}