

Ajax

布尔教育 <http://www.itbool.com>

燕十八 著

严禁传播 违者必究

第1节 Ajax原理

访问网站,离不开HTTP协议,离不开请求与响应.

Client----请求--->Server

Client<----响应----Server

问: 你见过哪些Client?

答: Firefox,chrome,等浏览器.

问: Client只能是浏览器吗?

答: 凡能发送HTTP请求的,都可称为Client

如下图:

```
POST /http/02.php HTTP/1.1
Host: localhost
Content-type: application/x-www-form-urlencoded
Content-length: 10

name=white
```

浏览器中,提供了一个JS对象--XMLHttpRequest

它也可以帮我们发送HTTP请求,并接受Server的响应.

这意味着,我们的浏览器不提交,通过JS也可以请求服务器.

Ajax,其实就是通过XHR对象,执行HTTP请求.

所以,学习的关键是---弄清XHR对象的属性和方法.

第2节 XHR对象

2.1 创建XHR对象?

var xhr = new XMLHttpRequest(); // 老旧IE有不同,不必考虑

2.2 对象的属性和方法

方法:

open('get/post',url, true/false); // 第3个参数先别管

send(null); //如有参数则k=v&k2=v2这种形式传过去,无参数则填null

属性

readyState: // 代表请求的状态,不断变化,0变化到4的时候,请求全部结束

responseText // 响应的内容

status // 响应状态码200,403,404

statusText // 状态文字,ok ,forbidden, not found

事件:

onreadystatechange=function(){} // 当readyState变化时,会触发此事件

第3节 Ajax验证用户名

```
<html>
```

```

<body>
  <p>用户名:<input type="text" name="username" /><span id="reg"></span>
</p>
</body>
<script>
document.getElementsByName('username')[0].onblur = function() {
  var xhr = new XMLHttpRequest();
  var url = '02.php?username=' + this.value;

  xhr.onreadystatechange = function(){
    if(xhr.readyState == 4) {
      alert('请求结束了,响应内容是' + this.responseText);
    }
  }

  xhr.open('GET' , url , true);
  xhr.send(null);
}
</script>
</html>

```

```

sleep(2);

$u = $_GET['username'];
$mysql = array('zhangsan' , 'lisi' , 'wangwu');

echo in_array($u, $mysql)?1:0;

```

当得到返回值0/1后,也可以通过DOM操作提示用户:

```

if(xhr.readyState == 4) {
  //alert('请求结束了,响应内容是' + this.responseText);
  var span = document.getElementById('reg');
  if(this.responseText == '0') {
    span.innerHTML = '<font color="green">恭喜,用户名可用</font>';
  } else {
    span.innerHTML = '<font color="red">可悲,用户名已被占用</font>';
  }
}

```

第4节 xhr对象如何发送POST查询?

关键点:

- open()的第1个参数用'POST'
- POST的参数以k1=v1&k2=v2&k3=v3的形式拼接,并用send()发送
- 必须要设置Content-Type,且值'application/x-www-form-urlencoded'

```

<body>
  <form action="">
    <p>用户名<input type="text" name="username" /></p>
    <p>密 码<input type="text" name="email"/></p>
    <p><input type="submit" value="注册" /></p>
  </form>
  <script>
    document.getElementsByTagName('form')[0].onsubmit = function() {
      var xhr = new XMLHttpRequest();
      var inps = document.getElementsByTagName('input');

      xhr.open('POST' , '04.php' , true);
      xhr.setRequestHeader('Content-Type' , 'application/x-www-form-urlencoded');
      xhr.send('username='+inps[0].value + '&email='+inps[1].value);
      return false;
    }
  </script>

```

```
</script>
</body>
```

第5节 返回值 json,html,text,XML 的区别？

返回值只有2种text, XML两种.

但text的内容可以是一段HTML,也可以是json结构的字符串.

5.1 返回JSON格式

应用场合: 多条或多种数据,需要清晰传递时

关键点: json格式的字符串,要eval一下,转为js的对象

```
<body>
  <input type="button" onclick="du();" value="随便看看">
  <h1></h1>
  <h2></h2>
  <p></p>
</body>
<script>
  function du() {
    var xhr = new XMLHttpRequest();

    xhr.open('get' , './05.php' , true);
    xhr.send(null);

    xhr.onreadystatechange = function() {
      if(this.readyState == 4) {
        //console.log(this.responseText);
        var poem = eval('(' + this.responseText + ')')
        console.log(poem);

        document.getElementsByTagName('h1')[0].innerHTML = poem.title;
        document.getElementsByTagName('h2')[0].innerHTML = poem.author;
        document.getElementsByTagName('p')[0].innerHTML = poem.content;
      }
    }
  }
</script>
```

```
$poem = array();

$poem[] = array('title'=>'渡汉江',
               'author'=>'李频', 'content'=>
               '岭外音书绝，经冬复立春。
               近乡情更怯，不敢问来人');

$poem[] = array('title'=>'寻隐者不遇',
               'author'=>'贾岛', 'content'=>
               '松下问童子，言师采药去。
               只在此山中，云深不知处。');

$poem[] = array('title'=>'登乐游原',
               'author'=>'李商隐', 'content'=>
               '向晚意不适，驱车登古原。
               夕阳无限好，只是近黄昏。');

$poem[] = array('title'=>'何满子',
               'author'=>'张祜', 'content'=>
               '故国三千里，深宫二十年。
               一声何满子，双泪落君前。');

$poem[] = array('title'=>'江雪',
               'author'=>'柳宗元', 'content'=>
               '千山鸟飞绝，万径人踪灭。
               孤舟蓑笠翁，独钓寒江雪。');

$i = mt_rand(0,4);
```

```
print_r($poem[$i]);
```

5.2 返回HTML代码

应用场合: 需要更新大面积HTML内容时.

```
header('Content-type:text/html; charset=utf-8');
$gn = array(
    '甲乙丙丁',
    '鸡毛蒜皮',
    '天南海北');
?>

<table>
<?php foreach($gn as $n) { ?>
<tr><td><?php echo $n;?></td></tr>
<?php } ?>
</table>
```

```
<body>
  <h1>直接返回大段HTML</h1>
  <a href="" onmouseover="news(1)">国内新闻</a>
  <a href="" onmouseover="news(2)">国际新闻</a>
  <div id="n">

  </div>
</body>
<script>
  function news(type) {
    var url = '06.php?n='+ type;
    var xhr = new XMLHttpRequest();

    xhr.onreadystatechange = function() {
      if(this.readyState == 4 && this.status == 200) {
        document.getElementById('n').innerHTML = this.responseText;
      }
    }

    xhr.open('GET' , url , true);
    xhr.send(null);

    return false;
  }
</script>
```

5.3 返回XML

应用场合: 返回大批量格式数据(已大部分被json取代)

关键点: 返回值用responseXML属性来读取

且responseXML属性不是字符串,而是DOM对象

```
<?php
sleep(3);
header('content-type: text/xml; charset=utf-8');
?>
<?xml version="1.0" encoding="utf-8"?>
<news><title>aaaa</title><title>bbbb</title><title>cccc</title></news>
```

```
<script>
var url = '07.php';
var xhr = new XMLHttpRequest();

xhr.onreadystatechange = function() {
  if(this.readyState == 4 && this.status == 200) {
```

```

        console.log( alert(this.responseXML) );
        var ts = this.responseXML.getElementsByTagName('title');
        alert(ts[0].childNodes[0].nodeValue);
    }
}

xhr.open('GET' , url , true);
xhr.send(null);
</script>

```

第6节 同步和异步的区别

AJAX即“Asynchronous Javascript + XML”（异步JavaScript和XML）

```

<script>
function t1 (){
    var xhr = new XMLHttpRequest();

    xhr.onreadystatechange = function() {
        if(this.readyState == 4) {
            alert(this.responseText);
        }
    }

    xhr.open('get' , '08.php' , true);
    xhr.send(null); //

    return this.responseText;
}

t1();
</script>
<body>
    <h1>异步的概念</h1>
    <p>异步不阻塞后面代码的执行</p>
</body>
<script>
function t2() {
    alert('t2 dang');
}
t2();
</script>

```

```

// 08.php
sleep(5);
echo 't1 dang';

```

第7节 Ajax跨域

ajax默认情况下只能请求同域名下的内容

HTML5

https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS

Access-Control-Allow-Origin: *

jsonp

jsonp严格来说,不属于ajax技术!

```

<body>
    <h1>360搜索推荐</h1>
    <p><input type="text" name="word" onkeyup="test1();" /></p>

```

```

    <ul></ul>
</body>
<script>
    function test1() {
        var inp = document.getElementsByTagName('input')[0];
        var url = 'http://sug.so.360.cn/suggest?callback=suggest_so&encodein=utf-8&encodeout=utf-8&format=json&fields=word,obdata&';
        //alert(url);

        var sc = document.createElement('script');
        sc.src = url;
        document.getElementsByTagName('head')[0].appendChild(sc);
    }

    function suggest_so(obj) {
        //alert(obj);
        var html = '';
        for(var i =0; i< obj.result.length; i+=1) {
            html = html + '<li>' + obj.result[i].word + '</li>';
        }

        document.getElementsByTagName('ul')[0].innerHTML = html;
    }
</script>

```

第8节 ajax上传文件

从http协议的角度看,上传文件,需要把文件的内容放在协议的主体信息里发送.

ajax上传文件,则意味着,js的XMLHttpRequest对象,可以获取文件的信息.

出于安全角度,JS不能获取磁盘上的文件.

因此, ajax上上传文件,做不到.

网上的ajax上传,往往是用swf,或iframe伪装,不是真的ajax上传.

但是,HTML5后,浏览器新增了FormData对象,且XHR对象也新增了一些功能.

可以让我们完成ajax上传,且有进度条效果.

```

<body>
    <form action="" method="post" enctype="multipart/form-data">
        <p><input type="text" name="username"></p>
        <p><input type="file" name="pic"></p>
        <p><input type="submit" value="走"></p>
        <div style="width:500px;height:10px;border:1px solid blue">
            <div id="prog" style="height:100%;width:0%;background:blue;"></div>
        </div>
    </form>
</body>
<script>
document.getElementsByTagName('form')[0].onsubmit = function() {
    var fd = new FormData(this);

    var xhr = new XMLHttpRequest();
    console.log(xhr);
    xhr.onreadystatechange = function() {
        if(this.readyState == 4) {
            alert(this.responseText);
        }
    }

    xhr.upload.onprogress = function (ev) {
        //console.log(ev);
        if(ev.lengthComputable==true) {
            var per = parseInt(100 * ev.loaded / ev.total);
            document.getElementById('prog').style.width = (per + '%');
        }
    }
}

```

```
}

// ajax有"同源策略",即只能请求相同域名下的URL
xhr.open('post' , '11.php' , true);
xhr.send(fd);

return false;
};
</script>
```

错误查找

半醉半醒 18:47:19

十八哥, 问你一下, 我遇到的问题。

ajax, 调取数据

数据用同一段php代码生成的

ie下, 获取为空,

google下, 却能获取成功,

<http://www.gogowon.com/say.php> (注,现已修复)

半醉半醒 18:48:33

给点建议吧, 我郁闷了, 大半天了

my的那个地址, 获取数据

提示: 中文参数 urlencoded

参考资料

MDN XMLHttpRequest <https://developer.mozilla.org/zh-CN/docs/Web/API/XMLHttpRequest>

从此不再惧怕URI编码 <http://www.cnblogs.com/artwl/archive/2012/03/07/2382848.html>