

Ajax

作者: 西岭

Email: xiling3105@163.com

指导: 燕十八

出自: 布尔教育--高级PHP工程师培训

注: 此文档不得转载, 不得用于商业用途, 不得修改转播;

如有其它用途或勘误, 请以邮件方式联系作者; 欢迎骚扰, 违者必究!!!

第1章、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章、XHR对象

2.1、创建XHR对象

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

2.2、对象的属性和方法

方法:

`open('get/post', URL, true/false);` // 第三个参数先别管

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

1.php

```
echo 11;
```

1.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <input type="submit" onclick="rep()" value="请求">
  </body>
  <script type="text/javascript">
    var inp = document.getElementsByTagName('input')[0];
    function rep(){
      var xhr = new XMLHttpRequest();
      // console.log(xhr);
      xhr.open('get', '1.php', true);
      xhr.send(null);
      xhr.onreadystatechange = function(){
        // console.log(xhr);
```

```

        if(this.readyState == 4){
            alert(this.responseText);
        }
    }
}
</script>
</html>

```

第3章、Ajax验证用户名

2.PHP:

```

<?php
    $op = $_GET['op'];
    //$op = $_POST['nu'];
    $arr = array('lisi','wangwu');
    echo in_array($op,$arr) ? 1 : 0;
?>

```

3.1、XHR对象发送GET请求

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
    </head>
    <body>
        <p>
            用户名:<input type="text" name="name" value=""><span id='rep
'></span>
        </p>
        <p>
            <input type="submit" value="提交">
        </p>
    </body>
    <script type="text/javascript">
        //获取DOM对象
        var users = document.getElementsByName('name')[0];
        //绑定焦点离开对象
        users.onblur = function(){
            //alert(this.value);
            var url='2.php?op='+this.value;//ajax访问地址
            var xhr = new XMLHttpRequest();
            xhr.open('get',url,true);

```

```

        xhr.send(null);
        var sp = document.getElementById('rep');
        //ajax回调函数
        xhr.onreadystatechange = function(){
            //判断请求状态
            if(this.readyState == 4){
                if(this.responseText == 1){
                    sp.innerHTML = 'yes';
                }
                else {
                    sp.innerHTML = 'no'
                }
            }
        }
    }
</script>
</html>

```

3.2、XHR对象发送POST请求

与get请求不同:

//打开连接

xhr.open ('post',path,true) ;

//设置请求头

xhr.setRequestHeader('Content-type','application/x-www-form-urlencoded');

//发送请求

xhr.send ('k1=v1&k2=v2&k3=v3') ;

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title></title>
</head>
<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>
</body>
<script type="text/javascript">
    var a = document.getElementsByTagName('input')[0];
    a.onblur = function (){
        var xhr = new XMLHttpRequest();
        var data = 'nu='+this.value+'&v=sa';
        xhr.open('POST','2.php',true);
        xhr.setRequestHeader('Content-type','application/x-www-form-url

```



```

encoded');
    xhr.send(data);
    return false;
}
</script>
</html>

```

第4章、返回值

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

json字符串转为JS的对象有两种方式: JSON.parse()、eval--不建议使用, 存在注入危险;

JSON.parse() 方法可以将一个 JSON 字符串解析成为一个 JavaScript 值。在解析过程中, 还可以选择性的篡改某些属性的原始解析值。

4.1、使用json

4.php

```

$arr = array('title' =>'jds', 'name'=>'123', 'user'=>'年合肥');
echo json_encode($arr);

```

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title></title>
</head>
<body>
</body>
<script type="text/javascript">
    var xhr = new XMLHttpRequest();
    xhr.open('get', '4.php', true);
    xhr.send(null);
    xhr.onreadystatechange = function () {
        if (this.readyState == 4) {
            var j = this.responseText;
//            alert(j);
//            直接将json字符串当到函数内部即可
//            var zhuan = JSON.parse(j);
//            eval('(' + 内容后加两个括号 + ')') 固定格式,

```

```

        var zhuan = eval('(' + j + ')');
        alert(zhuan.user);
    }
}
</script>
</html>

```

4.2、使用XML

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
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>

```

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title></title>
</head>
<body>
<input type="button" name="name" value="出来" onclick="kan()">
</body>
<script type="text/javascript">
    function kan(){
        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function(){
            if(this.readyState == 4){
                // alert(typeof(this.responseText)); //string
                // alert(this.responseXML); //xml对象
                var ins = this.responseXML.getElementsByTagName('title')
                [0];
                console.log(ins.childNodes[0].nodeValue);
            }
        }
        xhr.open('get','6.php',true);
        xhr.send(null);
    }
</script>
</html>

```

第5章、同步和异步

open的第三个参数

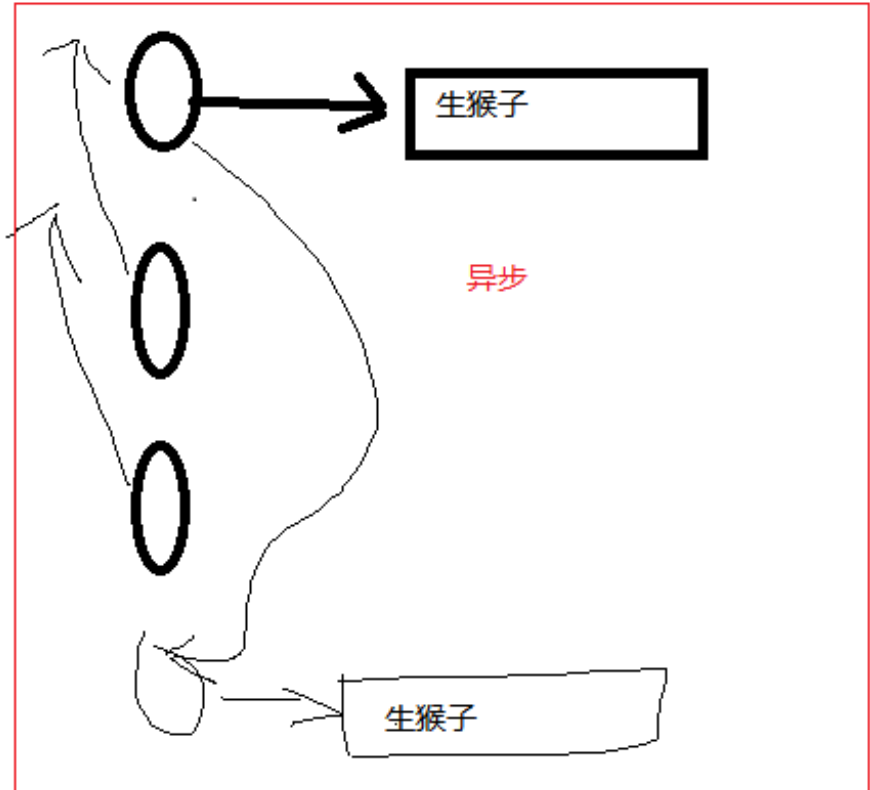
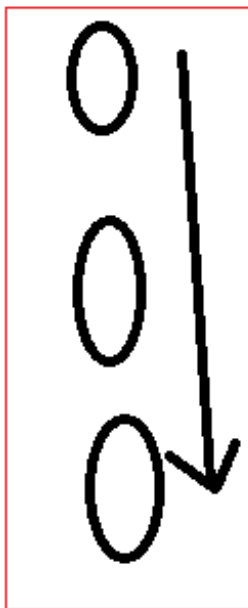
```
sleep(3);  
echo mt_rand(1,999);
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title></title>  
  </head>  
  <body>  
  
  </body>  
  <script type="text/javascript">  
    function t1(){  
      var xhr = new XMLHttpRequest();  
      xhr.onreadystatechange = function(){  
        if(this.readyState == 4){  
          alert(this.responseText);  
        }  
      }  
      xhr.open('get','7.php',false); //false阻塞 非异步//true非阻塞 异步  
      xhr.send(null);  
    }  
    function t3(){  
      alert('tttttt');  
    }  
    t1();  
    t3();  
  </script>  
</html>
```

图解同步和异步



同步



异步

第6章、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

php代码中添加一下header头声明:

Access-Control-Allow-Origin:* //域名, * 允许所有
能不能实现, 取决于你请求的这个地址;

php: (服务端代码)

shop.com中的PHP

```
<?php
```

```
header('Access-Control-Allow-Origin:http://localhost'); //允许此地址对我  
跨域请求
```

```
echo 1;
```



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<script type="text/javascript">
  function t1(){
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function(){
      if(this.readyState == 4){
        alert(this.responseText);
      }
    }
    xhr.open('get','http://shop.com',true); //true
    xhr.send(null);
  }
  t1();
</script>
</body>
</html>
```

第7章、利用jsonp的方式实现跨域访问

- 1: Ajax直接请求普通文件存在跨域无权限访问的问题，甭管你是静态页面、动态网页、web服务，只要是跨域请求，一律不准；
- 2: Web页面上调用js文件时则不受是否跨域的影响（不仅如此，我们还发现凡是拥有”src”这个属性的标签都拥有跨域的能力，比如script、img、iframe）；
- 3: 就是在远程服务器上设法动态的把数据装进js格式的文件里，供客户端调用和进一步处理；
- 4: 客户端获取数据后，按照自己的需求进行处理，这种方式看起来像ajax,但是和ajax并没有关系；
- 5: 为了便于客户端使用数据，逐渐形成了一种非正式传输协议，人们把它称作JSONP，该协议的一个要点就是允许用户传递一个callback参数给服务端，然后服务端返回数据时会将这个callback参数作为函数名来包裹住JSON数据，这样客户端就可以随意定制自己的函数来自动处理返回数据了。

例：利用搜索引擎展示jsonp数据格式的跨域

分析360搜索引擎的推举搜索推举过程；

10.js

```
so({"query":"f","result":[{"word":"fifa online3"}, {"word":"fedex"}, {"word":"fanc1"}, {"word":"fifa"}, {"word":"flash player官方下载"}, {"word":"flash"}, {"word":"facebook"}, {"word":"fc2"}, {"word":"fx"}, {"word":"ft中文网"}]}, "version":"3.2.1", "rec":"fifa online3");
```

10.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<script>
function so(test){
  console.log(test);
}
</script>
<script src="10.js"></script>
</body>
</html>
```

做一个搜索推举(没有网络可以直接忽略, 理解Jsonp的跨域原理即可)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <h1>360推举搜索</h1>
    <input type="text" name="name" value="" onkeyup="so()">
    <ul></ul>
  </body>
  <script type="text/javascript">
function so(){
  var inp = document.getElementsByTagName('input')[0];
  //将input的值添加到地址中
  var url = 'http://sug.so.360.cn/suggest?callback=suggest_so&encodein=utf-8&encodeout=utf-8&format=json&fields=word,obdata&word='+inp.value;
  //创建script标签, 并添加src属性
  var sc = document.createElement('script');
  sc.src = url;
```

```

//将创建好的script标签添加到head中
document.getElementsByTagName('head')[0].appendChild(sc);
//刚好实现suggest_so函数的调用，此函数在代码中已经准备好
}
function suggest_so(text){
    // alert(text);
    var html = '';
    //接受参数是个对象，有result属性和值；循环创建
    for(var i=0 ;i<text.result.length;i++){
        html = html + '<li>'+ text.result[i].word + '</li>';
    }
    document.getElementsByTagName('ul')[0].innerHTML = html;
}
</script>
</html>

```

第8章、ajax文件上传

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.

iframe伪装的文件上传

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form action="12.php" method="post" enctype="multipart/form-data" target="frm">
    <p><input type="file" name="pic"></p>
    <p><input type="submit" name="pic" value="tijiao"></p>
</form>
<iframe src="12.php" frameborder="0" name="frm"></iframe>
</body>
</html>

```

```

<?php
if($_FILES){

```



```

    echo '成功';
}
?>

<h1>上传</h1>

```

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.

JS:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form action="12.php" method="post" enctype="multipart/form-data" target="frm">
    <p><input type="file" name="pic"></p>
    <p><input type="submit" name="pic" value="tijiao"></p>
</form>
</body>
<script>
    var fom = document.getElementsByTagName('form')[0];
    fom.onsubmit = function () {
        var fmdata = new FormData(this); //FormData 会把表单的数据(包括文件流), 整体打包
        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function () {
            //回调函数待命.....
            if (this.readyState == 4) {
                alert(this.responseText);
            }
        }
        xhr.open('post', '12-2.php', true);
        xhr.send(fmdata);
        return false;
    }
</script>
</html>

```

php:

```

print_r($_FILES);
print_r($_POST);

```


修改php

```
if($_FILES){
    echo move_uploaded_file($_FILES['pic']['tmp_name'], './' . $_FILES['pic']['name'])?'ok':'dd';
}
```

注：到达此步，上传已经成功

修改HTML实现进度条效果

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<div style="width:500px;height:10px;border:1px solid blue">
    <div id="prog" style="height:100%;width:0%;background:blue;"></div>
</div>

<form action="12.php" method="post" enctype="multipart/form-data" target="frm">
    <p><input type="file" name="pic"></p>
    <p><input type="submit" name="pic" value="tijiao"></p>
</form>
</body>
<script>
    var fom = document.getElementsByTagName('form')[0];
    fom.onsubmit = function () {
        var fmdata = new FormData(this); //FormData 会把表单的数据(包括文件流)，整体打包
        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function () {
            // 回调函数待命
            if (this.readyState == 4) {
                alert(this.responseText);
            }
        }
        //打印上传进度
        //新的upload属性和过程事件onprogress
        xhr.upload.onprogress = function(ev){
            if(ev.lengthComputable){
                //打印进度信息
                // console.log(100 * ev.loaded/ev.total);
            }
        }
    }
}
```

```

        //将进度信息写入HTML
        var jd = 100 * ev.loaded/ev.total;
        document.getElementById('prog').style.width = jd+'%';
    }
}
xhr.open('post','12-2.php',true);
xhr.send(fmdata);
return false;
}
</script>
</html>

```

PHP:

```

<?php
// print_r($_FILES);
// print_r($_POST);
if($_FILES){
    echo move_uploaded_file($_FILES['pic']['tmp_name'], './' . $_FILES['pic']['name'])?'ok':'dd';
}

```

注意:

PHP.ini配置文件的上传限制

post_max_size=8M

upload_max_filesize=2M

第9章、URL的中文参数

encodeURIComponent

```

<script type="text/javascript">
    alert('布尔教育');
    alert(encodeURIComponent('布尔教育'));
</script>

```

扩展资料:

<http://www.cnblogs.com/artwl/archive/2012/03/07/2382848.html>

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