

# 学习笔记。。。积累！！

合抱之木,生于毫末。九层之台,起于累土。千里之行,始于足下

博客园 :: 首页 :: 博文 :: 闪存 :: 新随笔 :: 联系 :: 订阅  :: 管理 ::  85 随笔 :: 1 文章 :: 93 评论 :: 0 引用

< 2016年9月 >						
日	一	二	三	四	五	六
28	29	30	31	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	1
2	3	4	5	6	7	8

## 公告

### IT新闻:

昵称: [gaojun](#)  
园龄: 7年5个月  
粉丝: 260  
关注: 0  
[+加关注](#)

## 搜索

<input type="text"/>	<input type="button" value="找找看"/>
<input type="text"/>	<input type="button" value="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](#)

## 我的标签

[node.js\(2\)](#)  
[详解\(2\)](#)  
[远程连接\(1\)](#)  
[最大值\(1\)](#)  
[最小值\(1\)](#)  
[nodejs\(1\)](#)  
[oracle 10.2.0数据库。\(1\)](#)  
[pl/sql developer\(1\)](#)  
[process.nextTick\(1\)](#)  
[SVN\(1\)](#)  
[更多](#)

## 随笔分类

[C++\(6\)](#)  
[Firefox浏览器\(8\)](#)  
[JQuery\(2\)](#)  
[Linux\(12\)](#)  
[node.js\(3\)](#)  
[Web后台\(4\)](#)  
[Web前端\(21\)](#)

## XHR——XMLHttpRequest对象

# XHR——XMLHttpRequest对象

## 创建XMLHttpRequest对象

与之前众多DOM操作一样,创建XHR对象也具有兼容性问题:IE6及之前的版本使用ActiveXObject,IE7之后及其它浏览器使用XMLHttpRequest

不但IE6及其之前的版本将XHR作为一个ActiveXObject运行,而且还存在众多版本:一开始是Microsoft.XMLHTTP 之后变成Msxml2.XMLHTTP及更新版的Msxml3.XMLHTTP

```
function XHR() {  
    var xhr;  
    try {xhr = new XMLHttpRequest();}  
    catch(e) {  
        var IEXHRVers =  
        ["Msxml3.XMLHTTP", "Msxml2.XMLHTTP", "Microsoft.XMLHTTP"];  
        for (var i=0, len=IEXHRVers.length; i<  
len; i++) {  
            try {xhr = new  
ActiveXObject(IEXHRVers[i]);}  
            catch(e) {continue;}  
        }  
        return xhr;  
    }  
}
```

## 使用XHR

```
var xhr = XHR();  
//open方法 创建一个新的http请求,并指定此请求的方法、URL以及验证信息(用户名/密码)  
xhr.open("get", "test.txt", true);  
/*第一个参数是请求方式,一般用get与post方法,与form标签的method类似  
第二个参数是请求的URL  
第三个参数是请求是同步进行还是异步进行, true表示异步  
调用了open方法仅仅是传递了参数而已*/  
xhr.send(null); //调用了send方法后才会发出请求
```

## 杂项(3)

## 随笔档案

2014年12月 (5)  
2014年3月 (1)  
2014年1月 (2)  
2013年12月 (1)  
2013年11月 (1)  
2013年10月 (9)  
2013年6月 (1)  
2013年5月 (2)  
2013年4月 (1)  
2013年3月 (2)  
2013年1月 (1)  
2012年12月 (4)  
2012年11月 (3)  
2012年8月 (8)  
2012年7月 (1)  
2012年6月 (3)  
2012年5月 (4)  
2012年4月 (1)  
2010年9月 (2)  
2010年8月 (1)  
2010年5月 (1)  
2010年4月 (4)  
2010年3月 (2)  
2010年1月 (1)  
2009年12月 (2)  
2009年8月 (1)  
2009年7月 (2)  
2009年6月 (5)  
2009年5月 (11)  
2009年4月 (3)

## 相册

我(1)

## 最新评论

1. [Re:Node.js文件系统、路径的操作详解](#)  
很实用，谢谢。

--君小诺

2. [Re:jQuery extend方法使用及实现](#)  
膜拜大神~~~~!!!!

--study的猫

3. [Re:Visual Assist X设置](#)  
非常好啊。、。希望能翻译一下最新的。。现在助手设置里面变了很多

--长空X

4. [Re:Linux上Oracle 11g安装步骤图解](#)  
@鑫果果用root用户登录是不行的，要用oracle用户登录去执行才可以。...

--陈\_桂泉

## 阅读排行榜

1. [18种常见室内花卉的功效 \(转自网络\)\(155182\)](#)  
2. [Linux上Oracle 11g安装步骤图解\(148443\)](#)  
3. [查看Linux进程内存占用情况](#)

//并且get方式发送请求时send参数是null

在服务器环境中执行上面的脚本,并且给一个php或asp脚本发送请求,会发现服务器端脚本其实会被执行

```
//PHP脚本
$fp = fopen("a.txt", "wb");
fwrite($fp, "PHP文件在后台执行了");
fclose($fp);
echo "返回内容!";
```

上面PHP脚本的返回内容不会直接在页面上显示出来,必需要用JS通过XHR对象接收

```
var xhr = XHR();
xhr.open("get", "test.php", true);
xhr.onreadystatechange = callback;//在readystatechange事件
上绑定一个函数
//当接收到数据时,会调用readystatechange事件上的事件处理函
数
```

```
xhr.send(null);
function callback() {
    //在这里面没有使用this.readyState这是因为IE下面
ActiveXObject的特殊性
    if (xhr.readyState == 4) { //readyState表示文档加载
        进度, 4表示完毕
        alert(xhr.responseText); //responseText属性
        用来取得返回的文本
    }
}
```

## XHR对象参考

### readyState属性 返回当前请求的状态

- 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responseBody和responseText获取完整的回应数据

```
var xhr = XHR();
alert(xhr.readyState); //0
xhr.open("get", "test.htm", true);
alert(xhr.readyState); //1
xhr.send(null);
alert(xhr.readyState); //IE下会是4, 而FF下会是2
//可以通过readystatechange事件监听
xhr = XHR();
xhr.onreadystatechange = function () {
    alert(xhr.readyState); //FF下会依次是1, 2, 3, 4但最后
```

(146450)

[4. Linux LVM硬盘管理及LVM扩容 \(135708\)](#)[5. C++模板学习\(109661\)](#)

## 评论排行榜

[1. C++模板学习\(18\)](#)[2. Linux上Oracle 11g安装步骤图解\(14\)](#)[3. Linux LVM硬盘管理及LVM扩容 \(10\)](#)[4. 孙鑫视频VC++深入详解学习笔记\(10\)](#)[5. 借助网盘搭建SVN服务器\(7\)](#)

## 推荐排行榜

[1. C++模板学习\(36\)](#)[2. Linux上Oracle 11g安装步骤图解\(16\)](#)[3. Linux LVM硬盘管理及LVM扩容 \(13\)](#)[4. JS面向对象的程序设计\(9\)](#)[5. 孙鑫视频VC++深入详解学习笔记\(9\)](#)

还会再来个1

//IE下则是1, 1, 3, 4

};

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

xhr.send(null);

从上面可以看到,对于readyState这个属性,各个浏览器看法也不一样,但其实我们只需要知道当状态为4的时候可以获取response就行了!

**status 返回当前请求的http状态码**

status属性返回当前请求的http状态码,此属性仅当数据发送并接收完毕后才可获取。完整的HTTP状态码如下:

- 100 Continue 初始的请求已经接受, 客户应当继续发送请求的其余部分
- 101 Switching Protocols 服务器将遵从客户的请求转换到另外一种协议
- 200 OK 一切正常, 对GET和POST请求的应答文档跟在后面。
- 201 Created 服务器已经创建了文档, Location头给出了它的URL。
- 202 Accepted 已经接受请求, 但处理尚未完成。
- 203 Non-Authoritative Information 文档已经正常地返回, 但一些应答头可能不正确, 因为使用的是文档的拷贝
- 204 No Content 没有新文档, 浏览器应该继续显示原来的文档。如果用户定期地刷新页面, 而Servlet可以确定用户文档足够新, 这个状态代码是很有用的
- 205 Reset Content 没有新的内容, 但浏览器应该重置它所显示的内容。用来强制浏览器清除表单输入内容
- 206 Partial Content 客户发送了一个带有Range头的GET请求, 服务器完成了它
- 300 Multiple Choices 客户请求的文档可以在多个位置找到, 这些位置已经在返回的文档内列出。如果服务器要提出优先选择, 则应该在Location应答头指明。
- 301 Moved Permanently 客户请求的文档在其他地方, 新的URL在Location头中给出, 浏览器应该自动地访问新的URL。
- 302 Found 类似于301, 但新的URL应该被视为临时性的替代, 而不是永久性的。
- 303 See Other 类似于301/302, 不同之处在于, 如果原来的请求是POST, Location头指定的重定向目标文档应该通过GET提取
- 304 Not Modified 客户端有缓冲的文档并发出了一个条件性的请求 (一般是提供If-Modified-Since头表示客户只想比指定日期更新的文档)。服务器告诉客户, 原来缓冲的文档还可以继续使用。
- 305 Use Proxy 客户请求的文档应该通过Location头所指明的代理服务器提取
- 307 Temporary Redirect 和302 ( Found ) 相同。许多浏览器会错误地响应302应答进行重定向, 即使原来的请求是POST, 即使它实际上只能在POST请求的应答是303时才能重定向。由于这个原因, HTTP 1.1新增了307, 以便更加清除地区分几个状态代码: 当出现303应答时, 浏览器可以跟随重定向的GET和POST请求; 如果是307应答, 则浏览器只能跟随对GET请求的重定向。
- 400 Bad Request 请求出现语法错误。
- 401 Unauthorized 客户试图未经授权访问受密码保护的页面。应答

中会包含一个WWW-Authenticate头，浏览器据此显示用户名/密码对话框，然后在填写合适的Authorization头后再次发出请求。

- 403 Forbidden 资源不可用。
- 404 Not Found 无法找到指定位置的资源
- 405 Method Not Allowed 请求方法 ( GET、POST、HEAD、Delete、PUT、TRACE等 ) 对指定的资源不适用。
- 406 Not Acceptable 指定的资源已经找到，但它的MIME类型和客户在Accept头中所指定的不兼容
- 407 Proxy Authentication Required 类似于401，表示客户必须先经过代理服务器的授权。
- 408 Request Timeout 在服务器许可的等待时间内，客户一直没有发出任何请求。客户可以在以后重复同一请求。
- 409 Conflict 通常和PUT请求有关。由于请求和资源的当前状态相冲突，因此请求不能成功。
- 410 Gone 所请求的文档已经不再可用，而且服务器不知道应该重定向到哪一个地址。它和404的不同在于，返回407表示文档永久地离开了指定的位置，而404表示由于未知的原因文档不可用。
- 411 Length Required 服务器不能处理请求，除非客户发送一个Content-Length头
- 412 Precondition Failed 请求头中指定的一些前提条件失败
- 413 Request Entity Too Large 目标文档的大小超过服务器当前愿意处理的大小。如果服务器认为自己能够稍后再处理该请求，则应该提供一个Retry-After头
- 414 Request URI Too Long URI太长
- 416 Requested Range Not Satisfiable 服务器不能满足客户在请求中指定的Range头
- 500 Internal Server Error 服务器遇到了意料不到的情况，不能完成客户的请求
- 501 Not Implemented 服务器不支持实现请求所需要的功能。例如，客户发出了一个服务器不支持的PUT请求
- 502 Bad Gateway 服务器作为网关或者代理时，为了完成请求访问下一个服务器，但该服务器返回了非法的应答
- 503 Service Unavailable 服务器由于维护或者负载过重未能应答。例如，Servlet可能在数据库连接池已满的情况下返回503。服务器返回503时可以提供一个Retry-After头
- 504 Gateway Timeout 由作为代理或网关的服务器使用，表示不能及时地从远程服务器获得应答
- 505 HTTP Version Not Supported 服务器不支持请求中所指明的HTTP版本

事实上,我们只需要知道状态为200的时候(OK)才读取response就行了!

## responseText与responseXML

responseText 将响应信息作为字符串返回。XMLHTTP尝试将响应信息解码为Unicode字符串，XMLHTTP默认将响应数据的编码定为UTF-8，如果服务器返回的数据带BOM(byte-order mark)，XMLHTTP可以解码任何UCS-2 (big or little endian)或者UCS-4 数据。注意，如果服务器返回的是xml文档，此属性并不处理xml文档中的编码声明。你需要使用responseXML来处理。

responseXML 将响应信息格式化为Xml Document对象并返回。如果响应

数据不是有效的XML文档，此属性本身不返回XMLDOMParseError，可以通过处理过的DOMDocument对象获取错误信息。

## 其它一些XHR对象的方法

abort 取消当前请求

getAllResponseHeaders 获取响应的所有http头 每个http头名称和值用冒号分割，并以\r\n结束。当send方法完成后才可调用该方法。

getResponseHeader 从响应信息中获取指定的http头 当send方法成功后才可调用该方法。如果服务器返回的文档类型为"text/xml"，则这句话xmlhttp.getResponseHeader("Content-Type");将返回字符串"text/xml"。可以使用getAllResponseHeaders方法获取完整的http头信息。

setRequestHeader 单独指定请求的某个http头 如果已经存在已此名称命名的http头，则覆盖之。此方法必须在open方法后调用。

## 请求方式

### GET 请求

```
//JS
var xhr = XMLHttpRequest();
xhr.open("get", "test.php?
qs=true&userName=abc&pwd=123456", true);
xhr.onreadystatechange = function () {
    if (xhr.readyState==4 && xhr.status ==200) {
        alert(xhr.responseText);
    }
};
xhr.send(null);
//PHP
print_r($_GET);
```

### POST 请求

```
//JS
var xhr = XMLHttpRequest();
xhr.open("post", "test.php", true);
xhr.onreadystatechange = function () {
    if (xhr.readyState==4 && xhr.status ==200) {
        alert(xhr.responseText);
    }
};
//比GET请求多了一步
xhr.setRequestHeader("Content-Type", "application/x-www-
form-urlencoded");
//另外, 数据是通过send方法发送的
xhr.send("qs=true&userName=abc&pwd=123456");
//PHP
header("Cache-Control: no-cache, must-revalidate");//可以
让浏览器不缓存结果
```

```
print_r($_POST);
```

Email:gaojun\_le@163.com

分类: [Web前端](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

gaojun

关注 - 0

粉丝 - 260

[+加关注](#)

7

0

« 上一篇: [JGrouseDoc使用\(原创\)](#)

» 下一篇: [Ajax简介](#)

posted on 2012-08-11 18:47 [gaojun](#) 阅读(38809) 评论(1) [编辑](#) [收藏](#)

## 评论

### #1楼 2016-05-10 23:35 [codegay](#)

代码放到code标签中，有代码高亮有会好看很多。

[支持\(0\)](#) [反对\(0\)](#)

[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：

- [黑莓手机正式宣布“死亡”，它到底做错了什么？](#)
  - [Firefox Test Pilot项目迎来浮动视频和网页截图支持](#)
  - [Ubuntu 16.10（Yakkety Yak） Final Beta发布](#)
  - [新一代以太网标准获批 网速提高至少5倍](#)
  - [谷歌宣布Google Express服务将覆盖美国东北部](#)
- » [更多新闻...](#)

#### 最新知识库文章：

- [陈皓：什么是工程师文化？](#)
  - [没那么难，谈CSS的设计模式](#)
  - [程序猿媳妇儿注意事项](#)
  - [可是姑娘，你为什么要编程呢？](#)
  - [知其所以然（以算法学习为例）](#)
- » [更多知识库文章...](#)

Powered by:

