**Ajax手册**

1. [古代Ajax](#古代Ajax)
2. [现代Ajax](#现代Ajax)
3. [探讨Ajax](#探讨Ajax)
4. [HTML5与Ajax](#HTML5与Ajax)

五、[comet反向Ajax](#comet反向Ajax)

1. 古代Ajax
   1. 本质
   2. Iframe与局部刷新
   3. Js动态加载与无刷新请求
   4. 项目：无刷新投票
      1. 方法1：直接把投票结果写入到本地文件，利用’HTTP/1.1 204 No Content’请求不跳转
      2. 方法2：利用图片加载来请求
         1. Function vote(){
         2. var hh = document.createElement(‘img’);
         3. hh.setAttribute(‘src’,’./01-vote.php’);
         4. }
      3. 利用css,js加载，同上
      4. 方法4：利用iframe
         1. Width=”0” height=”0” frameBorder=”0”
      5. 所谓AJAX（JS的网络化）页面不刷新，利用XMLHttpRequest发送HTTP请求
   5. 以前的AJAX不能上传文件
   6. 总结：不使用XHR，用JS实现后台服务器请求，同时不刷新页面或跳转。
2. 现代Ajax
   1. XMLHttpRequest对象谈：
      1. 一个专门的HTTP请求工具
      2. Console.log(new XMLHttpRequest()); F12进入console控制台
      3. 创建对象
         1. 现代版本：new XMLHttpRequest()
         2. 低版本IE：new ActiveXObject('Microsoft.XMLHTTP')
         3. function createXHR(){
         4. var xhr = null;
         5. if(window.XMLHttpRequest){
         6. xhr = new XMLHttpRequest()
         7. }else if(window.ActiveXObject){
         8. xhr = new ActiveXObject('Microsoft.XMLHTTP');
         9. }
         10. return xhr;
         11. }
      4. 请求后台服务器资源
         1. 用什么方法请求：GET/POST/PUT/DELETE/HEAD
         2. 请求哪个资源： URL
         3. 同步方式 ： 同步/异步（false/true）
         4. function voteGet(){ **//GET请求**
         5. var xhr = createXHR();
         6. xhr.open(‘GET’,’./01-vote.php’,true); [\\true](file:///\\true):异步
         7. xhr.send(null);//参数k1=v1&k2=v2&kn=vn
         8. }
         9. function votePost(){ **//POST请求**
         10. var xhr = createXHR();
         11. xhr.open(‘POST’,’./02-vote.php’,true); [\\true](file:///\\true):异步
         12. xhr.onreadystatechange = function(){
         13. if(this.readyState=4){
         14. alert(this.responseTest);
         15. }
         16. }
         17. //收集表单数据
         18. var na = document.getElementByName(‘username’)[0].value;
         19. //设置请求头信息
         20. xhr.setRequestHeader(‘Content-Type’,’application/x-www-form-urlencoded’);
         21. xhr.send(‘username=’+na);//参数k1=v1&k2=v2&kn=vn
         22. }
      5. 请求的结果的利用
         1. alert(xhr.responseTest);
         2. 同步会等待返回值，储存在xhr.responseText上，成功send得到返回值
         3. 异步直接执行send后面语句，2秒之后xhr.responseText属性才有值
         4. xhr对象在请求与响应过程中，状态不断变化（0-4）逐步变化
            1. 绑定状态变化的回调函数
            2. xhr.onreadystatechange = function(){
            3. var pg = document.getElementById(‘progress’);
            4. pg.innerHTML = pg.innerHTML + ‘状态现在变成‘ + this.readyState + ‘<br />’;
            5. }
            6. 当readyState值为4时(最终状态)，请求完成；responseText储存返回值
            7. 当readyState值为0时，请求失败；
            8. 文档介绍

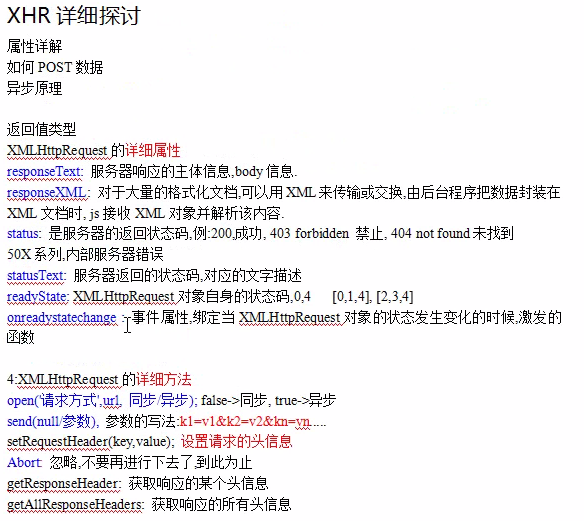
0 － （未初始化）还没有调用send()方法

1 － （载入）已调用send()方法，正在发送请求

2 － （载入完成）send()方法执行完成，已经接收到全部响应内容headers

3 － （交互）正在解析响应内容body

4 － （完成）响应内容解析完成，可以在客户端调用了



* 1. XMLHttpRequest兼容性
  2. 项目：更新天气预报

1. 探讨Ajax
   1. Ajax异步之原理
      1. 1.createXHR 2.open 3.onreadystatechange 4.send 5.alert
      2. 同步false：按顺序1，2，3，4
      3. 异步true：1，2，3，4，3（执行完才收到返回值）
   2. Ajax返回值之json与XML
      1. **返回值之XML**
      2. 需要XML源文档 头定义：（‘Content-Type: text/xml’）
      3. 返回对象var domxml = this.responseXML
      4. 即可以用DOM操作：var chs = domxml.getElementByTagName(‘book’)[0];
      5. alert(chs.firstChild.firstChild.wholeText);
      6. **返回值之JSON**
      7. 需要JSON源文档 {name:’zhangsan’,age:19}
      8. var user = eval(' ('+this.responseText+') ')
      9. alert(user.name);
   3. Ajax跨域之jsonp
      1. 百度JSONP：[http://suggestion.baidu.com/su?wd=#content#&cb=window.baidu.sug](http://suggestion.baidu.com/su?wd=#content)
      2. 项目：Ajax调用 百度JSONP 搜索结果
         1. **//Javascript代码**
         2. function mySearch(){
         3. var kw = document.getElementsByName('mykeywork')[0].value;
         4. var url = 'http://suggestion.baidu.com/su?wd='+kw+'&cb=mySearchFunction';
         5. var scr = document.createElement('script');
         6. scr.setAttribute('type','text/javascript');
         7. scr.setAttribute('src',url);
         8. document.getElementsByTagName('head')[0].appendChild(scr);
         9. }
         10. function mySearchFunction(src){
         11. //console.log(src);
         12. var searchWords = src.s;
         13. var l = document.getElementById('searchWordsList');
         14. var list = '';
         15. for(var i in searchWords){
         16. list += '<p>'+searchWords[i]+'</p>';
         17. }
         18. l.innerHTML = list;
         19. }
         20. **//HTML代码**
         21. <p>关键字：<input type="text" name="mykeywork"></p>
         22. <input type="button" value="搜索" onclick="mySearch();">
         23. <div id="searchWordsList"></div>
2. HTML5与Ajax
   1. XHR level2 新标准
   2. XHR2 操作二进制文件
   3. 项目：XHR2r拖拽上传2G超大文件
      1. file->继承自->Blob
      2. Blob有slice方法，可以截取二进制对象的一部分。
      3. while 还有数据｛
      4. 截取，
      5. Ajax上传
      6. }
   4. 上传文件方式
      1. iframe伪装，jquery-uploaded-file
         1. **script代码 (当前有一个loading.gif图片)**
         2. <script type="text/javascript" src="http://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"></script>
         3. /\*
         4. 1.捕捉表单提交动作
         5. 2.创建一个iframe
         6. 3.把表单的target修改指向该iframe
         7. 4.去掉这个iframe
         8. \*/
         9. function ajaxup(){
         10. var ifname = 'up'+Math.random();
         11. $('<iframe name="'+ifname+'" width="0" height="0" frameborder="0"></iframe>').appendTo($('body'));
         12. $('form:first').attr('target',ifname);
         13. $('#progress').html('<img src="./loading.gif" />');
         14. //return false;
         15. }
         16. **HTML代码**
         17. <h1>iframe模拟Ajax文件上传效果</h1>
         18. <h2></h2>
         19. <div id="progress"></div>
         20. <form action="http://localhost:8080/h5\_test/dist/up.jsp" method="post" enctype="multipart/form-data" onsubmit="return ajaxup();">
         21. <p><input type="file" name="pic"></p>
         22. <p><input type="submit" value="提交"></p>
         23. </form>
         24. **up.jsp代码（使用的域要一样localhost:8080）**
         25. //out.print("<script>parent.document.getElementsByTagName('h2')[0].innerHTML='succ'</script>");
         26. out.println("<script>parent.document.getElementById('progress').innerHTML='succ'</script>");
      2. swf插件
      3. html5
         1. XHR2 + FILE API + FormData
         2. **javascript代码**
         3. var fm = document.getElementById('tform');
         4. var fd = new FormData(fm);
         5. var xhr = new XMLHttpRequest();
         6. xhr.open('POST', './formDataTest.jsp',true);
         7. xhr.send(fd);
         8. //FormData可以从表单自动生成数据，也可以手动append数据
         9. fd.append('grade', '3');
         10. **FILE API 代码**
         11. HTML代码块
         12. <input type="file" name="pic" onchange="selfile();">
         13. javascript代码块
         14. function selfile(){
         15. var fd = new FormData();
         16. var pic = document.getElementsByTagName( 'input')[0].files[0];
         17. fd.append('pic',pic);
         18. ……
         19. }
         20. xhr……代码区同上
         21. 服务器端执行上传代码……xhr.open提交代码
         22. **二进制对象直接读成浏览器显示的资源**
         23. var tmpimg = document.createElement(‘img’);
         24. tmpimg.src = window.URL.createObjectURL(pic);
         25. document.getElementsByTagName(‘body’)[0].appendChild(tmpimg);
         26. **利用XHR2的新标准，为上传过程写一个监听函数**
         27. /\*
         28. 进度条：总大小，已上传大小
         29. 在上传过程，函数读取已上传/总大小，不断触发函数更新进度条
         30. \*/
         31. xhr.upload.onprogress = function (ev) {
         32. if(ev.lengthComputable){
         33. var persent = 100\*ev.loaded/ev.total;
         34. document.getElementById('bar').style.width = persent + '%';
         35. }
         36. }
3. comet反向Ajax
   1. 长连接与chunk传输
   2. Iframe与chunk交换数据
   3. 利用js通讯
   4. 项目：在线客服系统