**JQuery.Ajax之错误调试帮助信息**

下面是Jquery中AJAX参数详细列表：

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 描述 |
| **url** | String | (默认: 当前页地址) 发送请求的地址。 |
| **type** | String | (默认: "GET") 请求方式 ("POST" 或 "GET")， 默认为 "GET"。注意：其它 HTTP 请求方法，如 PUT 和 DELETE 也可以使用，但仅部分浏览器支持。 |
| **timeout** | Number | 设置请求超时时间（毫秒）。此设置将覆盖全局设置。 |
| **async** | Boolean | (默认: true) 默认设置下，所有请求均为异步请求。如果需要发送同步请求，请将此选项设置为 false。注意，同步请求将锁住浏览器，用户其它操作必须等待请求完成才可以执行。 |
| **beforeSend** | Function | 发送请求前可修改 XMLHttpRequest 对象的函数，如添加自定义 HTTP 头。XMLHttpRequest 对象是唯一的参数。  function (XMLHttpRequest) {  this; // the options for this ajax request } |
| **cache** | Boolean | (默认: true) jQuery 1.2 新功能，设置为 false 将不会从浏览器缓存中加载请求信息。 |
| **complete** | Function | 请求完成后回调函数 (请求成功或失败时均调用)。参数： XMLHttpRequest 对象，成功信息字符串。  function (XMLHttpRequest, textStatus) {  this; // the options for this ajax request } |
| **contentType** | String | (默认: "application/x-www-form-urlencoded") 发送信息至服务器时内容编码类型。默认值适合大多数应用场合。 |
| **data** | Object, String | 发送到服务器的数据。将自动转换为请求字符串格式。GET 请求中将附加在 URL 后。查看 processData 选项说明以禁止此自动转换。必须为 Key/Value 格式。如果为数组，jQuery 将自动为不同值对应同一个名称。如 {foo:["bar1", "bar2"]} 转换为 '&foo=bar1&foo=bar2'。 |
| **dataType** | String | 预期服务器返回的数据类型。如果不指定，jQuery 将自动根据 HTTP 包 MIME 信息返回 responseXML 或 responseText，并作为回调函数参数传递，可用值:  "xml": 返回 XML 文档，可用 jQuery 处理。  "html": 返回纯文本 HTML 信息；包含 script 元素。  "script": 返回纯文本 JavaScript 代码。不会自动缓存结果。  "json": 返回 JSON 数据 。  "jsonp": [JSONP](http://bob.pythonmac.org/archives/2005/12/05/remote-json-jsonp/) 格式。使用 [JSONP](http://bob.pythonmac.org/archives/2005/12/05/remote-json-jsonp/) 形式调用函数时，如 "myurl?callback=?" jQuery 将自动替换 ? 为正确的函数名，以执行回调函数。 |
| **error** | Function | (默认: 自动判断 (xml 或 html)) 请求失败时将调用此方法。这个方法有三个参数：XMLHttpRequest 对象，错误信息，（可能）捕获的错误对象。  function (XMLHttpRequest, textStatus, errorThrown) {  // 通常情况下textStatus和errorThown只有其中一个有值   this; // the options for this ajax request } |
| **global** | Boolean | (默认: true) 是否触发全局 AJAX 事件。设置为 false 将不会触发全局 AJAX 事件，如 ajaxStart 或 ajaxStop 。可用于控制不同的Ajax事件 |
| **ifModified** | Boolean | (默认: false) 仅在服务器数据改变时获取新数据。使用 HTTP 包 Last-Modified 头信息判断。 |
| **processData** | Boolean | (默认: true) 默认情况下，发送的数据将被转换为对象(技术上讲并非字符串) 以配合默认内容类型 "application/x-www-form-urlencoded"。如果要发送 DOM 树信息或其它不希望转换的信息，请设置为 false。 |
| **success** | Function | 请求成功后回调函数。这个方法有两个参数：服务器返回数据，返回状态  function (data, textStatus) {  // data could be xmlDoc, jsonObj, html, text, etc...  this; // the options for this ajax request } |

代码：$(document).ready(function() {

jQuery("#clearCac").click(function() {

jQuery.ajax({

url: "/Handle/Do.aspx",

type: "post",

data: { id: '0' },

dataType: "json",

success: function(msg) {

alert(msg);

},

error: function(XMLHttpRequest, textStatus, errorThrown) {

alert(XMLHttpRequest.status);

alert(XMLHttpRequest.readyState);

alert(textStatus);

},

complete: function(XMLHttpRequest, textStatus) {

this; // 调用本次AJAX请求时传递的options参数

}

});

});

});

一、error：function (XMLHttpRequest, textStatus, errorThrown)

{

}

(默认: 自动判断 (xml 或 html)) 请求失败时调用时间。参数有以下三个：XMLHttpRequest 对象、错误信息、（可选）捕获的错误对象。如果发生了错误，错误信息（第二个参数）除了得到null之外，还可能是"timeout", "error", "notmodified" 和 "parsererror"。

textStatus:

null, "timeout", "error", "notmodified" 和 "parsererror"。

二、error事件返回的第一个参数XMLHttpRequest有一些有用的信息：

XMLHttpRequest.readyState:

状态码

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

3 － （交互）正在解析响应内容

4 － （完成）响应内容解析完成，可以在客户端调用了

三、data:"{}", data为空也一定要传"{}"；不然返回的是xml格式的。并提示parsererror.

四、parsererror的异常和Header 类型也有关系。及编码header('Content-type: text/html; charset=utf8');

五、XMLHttpRequest.status:

1xx-信息提示

这些状态代码表示临时的响应。客户端在收到常规响应之前，应准备接收一个或多个1xx响应。

100-继续。

101-切换协议。

2xx-成功

这类状态代码表明服务器成功地接受了客户端请求。

200-确定。客户端请求已成功。

201-已创建。

202-已接受。

203-非权威性信息。

204-无内容。

205-重置内容。

206-部分内容。

3xx-重定向

客户端浏览器必须采取更多操作来实现请求。例如，浏览器可能不得不请求服务器上的不同的页面，或通过代理服务器重复该请求。

301-对象已永久移走，即永久重定向。

302-对象已临时移动。

304-未修改。

307-临时重定向。

4xx-客户端错误

发生错误，客户端似乎有问题。例如，客户端请求不存在的页面，客户端未提供有效的身份验证信息。400-错误的请求。

401-访问被拒绝。IIS定义了许多不同的401错误，它们指明更为具体的错误原因。这些具体的错误代码在浏览器中显示，但不在IIS日志中显示：

401.1-登录失败。

401.2-服务器配置导致登录失败。

401.3-由于ACL对资源的限制而未获得授权。

401.4-筛选器授权失败。

401.5-ISAPI/CGI应用程序授权失败。

401.7–访问被Web服务器上的URL授权策略拒绝。这个错误代码为IIS6.0所专用。

403-禁止访问：IIS定义了许多不同的403错误，它们指明更为具体的错误原因：

403.1-执行访问被禁止。

403.2-读访问被禁止。

403.3-写访问被禁止。

403.4-要求SSL。

403.5-要求SSL128。

403.6-IP地址被拒绝。

403.7-要求客户端证书。

403.8-站点访问被拒绝。

403.9-用户数过多。

403.10-配置无效。

403.11-密码更改。

403.12-拒绝访问映射表。

403.13-客户端证书被吊销。

403.14-拒绝目录列表。

403.15-超出客户端访问许可。

403.16-客户端证书不受信任或无效。

403.17-客户端证书已过期或尚未生效。

403.18-在当前的应用程序池中不能执行所请求的URL。这个错误代码为IIS6.0所专用。

403.19-不能为这个应用程序池中的客户端执行CGI。这个错误代码为IIS6.0所专用。

403.20-Passport登录失败。这个错误代码为IIS6.0所专用。

404-未找到。

404.0-（无）–没有找到文件或目录。

404.1-无法在所请求的端口上访问Web站点。

404.2-Web服务扩展锁定策略阻止本请求。

404.3-MIME映射策略阻止本请求。

405-用来访问本页面的HTTP谓词不被允许（方法不被允许）

406-客户端浏览器不接受所请求页面的MIME类型。

407-要求进行代理身份验证。

412-前提条件失败。

413–请求实体太大。

414-请求URI太长。

415–不支持的媒体类型。

416–所请求的范围无法满足。

417–执行失败。

423–锁定的错误。

5xx-服务器错误

服务器由于遇到错误而不能完成该请求。

500-内部服务器错误。

500.12-应用程序正忙于在Web服务器上重新启动。

500.13-Web服务器太忙。

500.15-不允许直接请求Global.asa。

500.16–UNC授权凭据不正确。这个错误代码为IIS6.0所专用。

500.18–URL授权存储不能打开。这个错误代码为IIS6.0所专用。

500.100-内部ASP错误。

501-页眉值指定了未实现的配置。

502-Web服务器用作网关或代理服务器时收到了无效响应。

502.1-CGI应用程序超时。

502.2-CGI应用程序出错。application.

503-服务不可用。这个错误代码为IIS6.0所专用。

504-网关超时。

505-HTTP版本不受支持。

FTP

1xx-肯定的初步答复

这些状态代码指示一项操作已经成功开始，但客户端希望在继续操作新命令前得到另一个答复。

110重新启动标记答复。

120服务已就绪，在nnn分钟后开始。

125数据连接已打开，正在开始传输。

150文件状态正常，准备打开数据连接。

2xx-肯定的完成答复

一项操作已经成功完成。客户端可以执行新命令。200命令确定。

202未执行命令，站点上的命令过多。

211系统状态，或系统帮助答复。

212目录状态。

213文件状态。

214帮助消息。

215NAME系统类型，其中，NAME是AssignedNumbers文档中所列的正式系统名称。

220服务就绪，可以执行新用户的请求。

221服务关闭控制连接。如果适当，请注销。

225数据连接打开，没有进行中的传输。

226关闭数据连接。请求的文件操作已成功（例如，传输文件或放弃文件）。

227进入被动模式(h1,h2,h3,h4,p1,p2)。

230用户已登录，继续进行。

250请求的文件操作正确，已完成。

257已创建“PATHNAME”。

3xx-肯定的中间答复

该命令已成功，但服务器需要更多来自客户端的信息以完成对请求的处理。331用户名正确，需要密码。

332需要登录帐户。

350请求的文件操作正在等待进一步的信息。

4xx-瞬态否定的完成答复

该命令不成功，但错误是暂时的。如果客户端重试命令，可能会执行成功。421服务不可用，正在关闭控制连接。如果服务确定它必须关闭，将向任何命令发送这一应答。

425无法打开数据连接。

426Connectionclosed;transferaborted.

450未执行请求的文件操作。文件不可用（例如，文件繁忙）。

451请求的操作异常终止：正在处理本地错误。

452未执行请求的操作。系统存储空间不够。

5xx-永久性否定的完成答复

该命令不成功，错误是永久性的。如果客户端重试命令，将再次出现同样的错误。500语法错误，命令无法识别。这可能包括诸如命令行太长之类的错误。

501在参数中有语法错误。

502未执行命令。

503错误的命令序列。

504未执行该参数的命令。

530未登录。

532存储文件需要帐户。

550未执行请求的操作。文件不可用（例如，未找到文件，没有访问权限）。

551请求的操作异常终止：未知的页面类型。

552请求的文件操作异常终止：超出存储分配（对于当前目录或数据集）。

553未执行请求的操作。不允许的文件名。

常见的FTP状态代码及其原因

150-FTP使用两个端口：21用于发送命令，20用于发送数据。状态代码150表示服务器准备在端口20上打开新连接，发送一些数据。

226-命令在端口20上打开数据连接以执行操作，如传输文件。该操作成功完成，数据连接已关闭。

230-客户端发送正确的密码后，显示该状态代码。它表示用户已成功登录。

331-客户端发送用户名后，显示该状态代码。无论所提供的用户名是否为系统中的有效帐户，都将显示该状态代码。

426-命令打开数据连接以执行操作，但该操作已被取消，数据连接已关闭。

530-该状态代码表示用户无法登录，因为用户名和密码组合无效。如果使用某个用户帐户登录，可能键入错误的用户名或密码，也可能选择只允许匿名访问。如果使用匿名帐户登录，IIS的配置可能拒绝匿名访问。

550-命令未被执行，因为指定的文件不可用。例如，要GET的文件并不存在，或试图将文件PUT到您没有写入权限的目录。