Ajax

* console.log打印返回结果，在网页控制台Console里查看。Network查看请求信息。
* contentType会影响返回结果，特别是post。
* 如若打印结果出现很多html语句，将语句复制到html文档里，能看见后台报错。
* 报错时执行error函数，返回错误代码和类型。
* php后台返回json类型，exit(json\_encode($params))

前端代码

|  |
| --- |
| <**script type="text/javascript"**>  **$**(***document***).ready(**function**(){  **$**(**'#w0'**).click(**function** () {  **$**.ajax({  **type**: **"GET"**,  **url**: **"{php echo webUrl('shop/test/ajax')}"**,  **contentType**:**"application/x-www-form-urlencoded"**,*//后台可获post参数  // contentType: 'text/html; charset=utf-8',//后台不可获post参数  // contentType: 'application/json; charset=utf-8',//后台不可获post参数* **data**: {**username**:**$**(**'#username'**).val(),**gender**:**$**(**'#gender'**).val()},  **dataType**: **"json"**,*//状态：json，text,string* success: **function** (data) {  *alert*(data.**result**.**username**+**'，'**+data.**result**.**gender**);  ***console***.log(data);  },  error: **function** (XMLHttpRequest, textStatus, errorThrown) {  *// 状态码* **$**(**".status"**).html(**'status:'**+XMLHttpRequest.**status**);  *// 状态* **$**(**".readyState"**).html(**'readyState:'**+XMLHttpRequest.**readyState**);  *// 错误信息* **$**(**".textStatus"**).html(**'textStatus:'**+XMLHttpRequest.**textStatus**);  }  });  })  }); </**script**> |

后台代码

|  |
| --- |
| **public function** ajax(){  $username = **'姓名：'**.$\_POST[**'username'**];  $gender =**'性别：'**.$\_POST[**'gender'**];  $this->show\_json1(1,**array**(**'username'**=>$username,**'gender'**=>$gender)); } **function** show\_json1($status = 1, $return = **NULL**) {  $ret = **array**( **"status"** => $status, **"result"** => ($status == 1 ? **array**( **"url"** => referer() ) : **array**( )) );  **if**( !*is\_array*($return) )  {  **if**( $return )  {  $ret[**"result"**][**"message"**] = $return;  }  **exit**( *json\_encode*($ret) );  }  $ret[**"result"**] = $return;  **if**( **isset**($return[**"url"**]) )  {  $ret[**"result"**][**"url"**] = $return[**"url"**];  }  **else** {  **if**( $status == 1 )  {  $ret[**"result"**][**"url"**] = referer();  }  }  **exit**( *json\_encode*($ret) ); } |

XMLHttpRequest

XMLHttpRequest对象

属性

|  |  |
| --- | --- |
| readyState | 值从 0 开始，直到接收到完整的 HTTP 响应，这个值增加到 4。每次这个属性的值增加的时候，都会触发 onreadystatechange 事件句柄。  0: 请求未初始化  1: 服务器连接已建立  2: 请求已接收  3: 请求处理中  4: 请求已完成，且响应已就绪 |
| responseText | 目前为止为服务器接收到的响应体（不包括头部），或者如果还没有接收到数据的话，就是空字符串。 |
| responseXML | 对xml文件请求的响应，解析为 XML 并作为 Document 对象返回。 |
| status | 由服务器返回的 HTTP 状态代码。当 readyState 小于 3 的时候读取这一属性会导致一个异常。  200: "OK"  404: 未找到页面 |
| statusText | HTTP 的状态名称 |

事件句柄

|  |  |
| --- | --- |
| onreadystatechange | 每次 readyState 属性改变的时候调用的事件句柄函数。当 readyState 为 3 时，它也可能调用多次。 |

方法

|  |  |
| --- | --- |
| abort() | 取消当前响应，关闭连接并且结束任何未决的网络活动。  这个方法把 XMLHttpRequest 对象重置为 readyState 为 0 的状态，并且取消所有未决的网络活动。例如，如果请求用了太长时间，而且响应不再必要的时候，可以调用这个方法。 |
| getAllResponseHeaders() | 把 HTTP 响应头部作为未解析的字符串返回。  如果 readyState 小于 3，这个方法返回 null。否则，它返回服务器发送的所有 HTTP 响应的头部。头部作为单个的字符串返回，一行一个头部。每行用换行符 "\r\n" 隔开。 |
| getResponseHeader() | 返回指定的 HTTP 响应头部的值。其参数是要返回的 HTTP 响应头部的名称。可以使用任何大小写来制定这个头部名字，和响应头部的比较是不区分大小写的。  该方法的返回值是指定的 HTTP 响应头部的值，如果没有接收到这个头部或者 readyState 小于 3 则为空字符串。如果接收到多个有指定名称的头部，这个头部的值被连接起来并返回，使用逗号和空格分隔开各个头部的值。 |
| open(method, url, async, username, password) | 初始化 HTTP 请求参数，例如 URL 和 HTTP 方法，但是并不发送请求。 |
| send(body) | 发送 HTTP 请求，使用传递给 open() 方法的参数，以及传递给该方法的可选请求体。 |
| setRequestHeader() | 向一个打开但未发送的请求设置或添加一个 HTTP 请求。 |

例子

ajax.html

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Title</**title**>  <**script**>  **function** *showHint*(str)  {  **if** (str.**length**==0)  {  ***document***.getElementById(**"txtHint"**).**innerHTML**=**""**;  **return**;  }  **if** (***window***.***XMLHttpRequest***)  {*// code for IE7+, Firefox, Chrome, Opera, Safari* **xmlhttp**=**new *XMLHttpRequest***();  }  **else** {*// code for IE6, IE5，低版本IE* **xmlhttp**=**new *ActiveXObject***(**"Microsoft.XMLHTTP"**);  }  **xmlhttp**.**onreadystatechange**=**function**()  {  **if** (**xmlhttp**.**readyState**==4 && **xmlhttp**.**status**==200)  {  ***document***.getElementById(**"txtHint"**).**innerHTML**=**xmlhttp**.**responseText**;  ***console***.log(**xmlhttp**.**responseXML**);  }  }  **xmlhttp**.open(**"GET"**,**"../controller/gethint.php?q="**+str,**true**);  *// xmlhttp.open("GET","../data/xml/test.xml",true);//返回xmlhttp.responseXML  // xmlhttp.open("GET","../welcome.txt",true);//返回文本文件全部内容* **xmlhttp**.send();  }  </**script**> </**head**> <**body**> <**p**><**b**>在输入框中输入一个姓名:</**b**></**p**> <**form**>  姓名: <**input type="text" onkeyup="***showHint*(**this**.**value**)**"**> </**form**> <**p**>返回值: <**span id="txtHint"**></**span**></**p**> </**body**> </**html**> |

gethint.php

|  |
| --- |
| **<?php** *// 将姓名填充到数组中* $a[]=**"Anna"**; $a[]=**"Brittany"**; $a[]=**"Cinderella"**; $a[]=**"Diana"**; $a[]=**"Eva"**; $a[]=**"Fiona"**; $a[]=**"Gunda"**; $a[]=**"Hege"**; $a[]=**"Inga"**; $a[]=**"Johanna"**; $a[]=**"Kitty"**; $a[]=**"Linda"**; $a[]=**"Nina"**; $a[]=**"Ophelia"**; $a[]=**"Petunia"**; $a[]=**"Amanda"**; $a[]=**"Raquel"**; $a[]=**"Cindy"**; $a[]=**"Doris"**; $a[]=**"Eve"**; $a[]=**"Evita"**; $a[]=**"Sunniva"**; $a[]=**"Tove"**; $a[]=**"Unni"**; $a[]=**"Violet"**; $a[]=**"Liza"**; $a[]=**"Elizabeth"**; $a[]=**"Ellen"**; $a[]=**"Wenche"**; $a[]=**"Vicky"**;  *//从请求URL地址中获取 q 参数* $q=$\_GET[**"q"**];  *//查找是否由匹配值， 如果 q>0* **if** (*strlen*($q) > 0) {  $hint=**""**;  **for**($i=0; $i<*count*($a); $i++)  {  **if** (*strtolower*($q)==*strtolower*(*substr*($a[$i],0,*strlen*($q))))  {  **if** ($hint==**""**)  {  $hint=$a[$i];  }  **else** {  $hint=$hint.**" , "**.$a[$i];  }  }  } }  *// 如果没有匹配值设置输出为 "no suggestion" // or to the correct values* **if** ($hint == **""**) {  $response=**"no suggestion"**; } **else** {  $response=$hint; }  *//输出返回值* **echo** $response; **?>** |

解释：ajax.html文件的input标签将输入此标签的值传到JavaScript函数showHint中。如果向函数showHint传入了任何文本（即 strlen($q) > 0），则会发生：

查找匹配 JavaScript 发送的字符的姓名

如果未找到匹配，则将响应字符串设置为 "no suggestion"

如果找到一个或多个匹配姓名，则用所有姓名设置响应字符串

把响应发送到 "txtHint" 占位符