# Ajax提交数据到后台

Ajax 即“**A***synchronous***J***avascript****A****nd***X***ML*”（异步 JavaScript 和 XML），是指一种创建交互式[网页](https://baike.baidu.com/item/%E7%BD%91%E9%A1%B5/99347)应用的网页开发技术。

Ajax 是一种在无需重新加载整个网页的情况下，能够更新部分网页内容的技术，用户体验非常好。

## ajax局部刷新技术



单击“ajax请求”按钮向后台请求数据，然后更新最新数据到上面的黄色区域。页面其他地方不会刷新，即整个页面不会有闪烁的现象。

## 利用jquery库提交ajax请求

**$**.**ajax**({  
 **type**:**"POST"**,  
 [**url**:**"/getUser**](url:%22/getUser)**"**,**data**:{  
 **id**: **'8'** }, *// 发送到服务器的数据。* **dataType**: **"json"**, *// 预期服务器返回的数据类型。* success:**function**(data){  
 **if** (data.ret == 1){  
 **$**(**'#userInfo'**).**html**(**'姓名:'** + data.detail.**name** + **'<br/>年龄:'** + data.detail.age);  
 }  
 **else**{  
 *alert*(**'获取数据失败!'**);  
 }  
 },  
 error: **function** (XMLHttpRequest, textStatus, errorThrown) {  
 *// 错误信息  
 alert*(textStatus);  
 }  
 });

（1）、要包含jquery库。

<script src="/static/lib/jquery\_3.3.1/jquery-3.3.1.min.js"></script>

（2）、更多的参数

**timeout**:   
要求为Number类型的参数，设置请求超时时间（毫秒）。此设置将覆盖$.ajaxSetup()方法的全局设置。

**async**:   
要求为Boolean类型的参数，默认设置为true，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，用户其他操作必须等待请求完成才可以执行。

**cache**:   
要求为Boolean类型的参数，默认为true（当dataType为script时，默认为false），设置为false将不会从浏览器缓存中加载请求信息。

**dataType**:   
要求为String类型的参数，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。可用的类型如下：  
xml：返回XML文档，可用JQuery处理。  
html：返回纯文本HTML信息；包含的script标签会在插入DOM时执行。  
script：返回纯文本JavaScript代码。不会自动缓存结果。除非设置了cache参数。注意在远程请求时（不在同一个域下），所有post请求都将转为get请求。  
json：返回JSON数据。  
jsonp：JSONP格式。使用SONP形式调用函数时，例如myurl?callback=?，JQuery将自动替换后一个“?”为正确的函数名，以执行回调函数。  
text：返回纯文本字符串。

**beforeSend**：  
要求为Function类型的参数，发送请求前可以修改XMLHttpRequest对象的函数，例如添加自定义HTTP头。在beforeSend中如果返回false可以取消本次ajax请求。XMLHttpRequest对象是惟一的参数。  
            function(XMLHttpRequest){  
               this;   //调用本次ajax请求时传递的options参数  
            }  
**complete**：  
要求为Function类型的参数，请求完成后调用的回调函数（请求成功或失败时均调用）。参数：XMLHttpRequest对象和一个描述成功请求类型的字符串。  
          function(XMLHttpRequest, textStatus){  
             this;    //调用本次ajax请求时传递的options参数  
          }

（2）、后台实例代码

@app.route(**'/getUser'**, methods=[**'POST'**])  
**def** getUser():  
 id = request.form[**'id'**]  
  
 *# 通常情况这里要根据id读取数据库得到用户的数据，这里仅仅模拟数据* userInfo = {  
 **'name'**:**'张三'**,  
 **'age'**:**'20'** }  
  
 ret = jsonify(ret=1, detail = userInfo)  
 **return** ret

访问效果：

