# Ajax

就是无需重新加载整个网页的情况下，才能够更新部分网页的技术

协议 域名 端口

[http://192.168.0.61:8080](http://192.168.0.61:8080/" \t "http://blog.csdn.net/iamlegendary/article/details/_blank)

## 同源策略

就是协议、域名、端口、都相同就是同源

$.ajax({

type:'get', 请求方式

url:'http://tom.com/jsonp.php', 地址

dataType:'jsonp', 返回数据，jsonp表示跨域请求数据

jsonp:'cb',//jsonp属性的作用就是自定义参数名字（callback=abc 这里的名字指的是 等号前面的键，后端根据这个键获取方法名，jquery的默认参数名称是callback）

jsonpCallback:'abc',//这个属性的作用就是自定义回调函数的名字 （callback=abc ，这里的名字指的是等号后面的值）

data:{}, 数据传到服务器的数据参数

success:function(data){ 服务器返回的数据

}

});

## Ajax请求流程

1 创建函数对象XMLHttpRuquest，通过这个对象向服务器发送请求，从服务器获取数据

2 给这个对象中的open（“get”，“url”“异步”）方法添加请求方式和地址

3 然后通过send( )发送 get请求 添加null,post请求，传递参数

4 指定回调函数

会触发.noreadystatechange( )事件处理程序

然后判断readystate ==4 status ==200 如果符合，就会返回数据通过

Readystate =0 为初始化 1发送请求 2浏览器收到服务器响应 3正在解析 4数据解析完成可以使用

responseXml和responseText获取完整的回应数据

function ajax4Json(defaults){  
 *// 1、创建XMLHttpRequest对象* var xhr = null;  
 if(**window**.XMLHttpRequest){  
 xhr = new XMLHttpRequest();  
 }else{  
 xhr = new ActiveXObject('Microsoft.XMLHTTP');  
 }  
 *// 把对象形式的参数转化为字符串形式的参数  
 /\*  
 {username:'zhangsan','password':123}  
 转换为  
 username=zhangsan&password=123  
 \*/* var param = '';  
 for(var attr in defaults.**data**){  
 param += attr + '=' + defaults.**data**[attr] + '&';  
 }  
 if(param){  
 param = param.substring(0,param.**length** - 1);  
 }  
 *// 处理get请求参数并且处理中文乱码问题* if(defaults.**type** == 'get'){  
 defaults.**url** += '?' + encodeURI(param);  
 }  
 *// 2、准备发送（设置发送的参数）* xhr.open(defaults.**type**,defaults.**url**,defaults.**async**);  
 *// 处理post请求参数并且设置请求头信息（必须设置）* var data = null;  
 if(defaults.**type** == 'post'){  
 data = param;  
 xhr.setRequestHeader("Content-Type","application/x-www-form-urlencoded");  
 }  
 *// 3、执行发送动作* xhr.send(data);  
 *// 处理同步请求，不会调用回调函数* if(!defaults.**async**){  
 if(defaults.**dataType** == 'json'){  
 **return JSON.parse(xhr.responseText);** }else{  
 **return xhr.responseText;** }  
 }  
 *// 4、指定回调函数（处理服务器响应数据）* xhr.onreadystatechange = function(){  
 if(xhr.**readyState** == 4){  
 if(xhr.**status** == 200){  
 var data = xhr.**responseText**;  
 if(defaults.**dataType** == 'json'){  
 *// data = eval("("+ data +")");* data = **JSON**.parse(data);  
 }  
 defaults.success(data);  
 }  
 }  
 }  
**}**

## ajax所包含的技术     1 使用CSS和XHTML来表示     2 使用DOM模型来交互和动态显示     3 使用XMLHttpRequest来和服务器进行异步通信     4 使用JavaScript来绑定和调用

Xml html js

## Ajax的优缺点

**Ajax的优点：**  
    1、 最大的一点是页面无刷新，在页面内与服务器通信，给用户的体验非常好。  
    2、 使用异步方式与服务器通信，不需要打断用户的操作，具有更加迅速的相应能力。  
    3、 可以把从前一些服务器负担的工作转嫁到客户端，利用客户端限制的能力来处理， 减轻服务器和带宽的负担，节约空间和宽带租用成本，并且减轻服务器的负担，Ajax 的原则是"按需取数据"，可以最大承德的减少冗余的请求和相应对服务器造成的负担。  
    4、 基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

**Ajax的缺：**

1 ajax破坏了后退按钮，破坏浏览器机制。

2 不安全，会暴露数据

3 对浏览器搜索引擎支持较弱

4 一些手持设备（如手机、PDA等）现在还不能很好的支持ajax，

5 破坏了程序的异常机制

## Josn数据后台的一种储存数据的类型

1 json数据没有变量

2 json数据的结尾没有分号

3 json中的数据必须有双引号的包裹

JSON 比 XML 更小、更快，更易解析

json.stringify( ) 把对象转换成 json字符串

json,parse( ） 把json字符串转换成对象

将数组和对象转换为字符串的方法 --- json\_encode();

将字符串转换为对象的方式 --- json\_decode();

## 异步效果与事件的处理机制

单线程 + 事件队列

事件队列中的任务执行的条件

1 主线已经空闲

2 任务满足触发条件

## 跨域

跨域解决方案

jsonp 99%的公司都在用

Iframe

跨域原理

动态创建script标签，（没有跨域限制的漏洞）通过src发出请求，服务器响应返回内容（函数调用）

dataType:”jsonp”

Jsonp:”回调函数的参数名字”

Callback:”callback函数”

跨域的缺点：

1 必须符合加载顺序

2 不方便通过程序传递参数

function ajax4Jsonp(defaults){  
 *// 这里是默认的回调函数名称  
 // expando: "jQuery" + ( version + Math.random() ).replace( /\D/g, "" ),* var cbName = 'jQuery' + ('1.11.1' + **Math**.random()).replace(/\D/g,"") + '\_' + (new Date().getTime());  
 if(defaults.jsonpCallback){  
 cbName = defaults.jsonpCallback;  
 }  
  
 *// 这里就是回调函数，调用方式：服务器响应内容来调用  
 // 向window对象中添加了一个方法，方法名称是变量cbName的值* **window**[cbName] = function(data){  
 defaults.success(data);*//这里success的data是实参* }  
  
 var param = '';  
 for(var attr in defaults.**data**){  
 param += attr + '=' + defaults.**data**[attr] + '&';  
 }  
 if(param){  
 param = param.substring(0,param.**length**-1);  
 param = '&' + param;  
 }  
 var script = **document**.createElement('script');  
 script.**src** = defaults.**url** + '?' + defaults.**jsonp** + '=' + cbName + param;  
 var head = **document**.getElementsByTagName('head')[0];  
 head.appendChild(script);  
**}**