Ajax 的全面总结

**一.什么是Ajax**

Ajax(Asynchronous JavaScript and XML),可以理解为JavaScript执行异步网络请求。通俗的理解的话就是，如果没有Ajax技术，改变网页的一小部分（哪怕是一行文字、一张图片）都需要重新加载一次整个页面，**而有了Ajax之后，就可以实现在网页不跳转不刷新的情况下，在网页后台提交数据，部分更新页面内容。**

**二.Ajax的原生写法**

**1.XMLHttpRequest对象**

XMLHttpRequest 对象用于在后台与服务器交换数据，能够在不重新加载页面的情况下更新网页，在页面已加载后从服务器请求数据，在页面已加载后从服务器接收数据，在后台向服务器发送数据。所以**XMLHttpRequest对象是Ajax技术的核心所在。**

**2.实现流程**

创建 XMLHttpRequest对象——>打开请求地址，初始化数据——>发送请求数据——>监听回调函数状态——>收到服务器返回的应答结果。

下面用具体的代码进行解释：

1. var xmlhttp;
2. function loadXMLDoc(url)
3. {
4. xmlhttp=null;
5. if (window.XMLHttpRequest)
6. {// code for all new browsers
7. xmlhttp=new XMLHttpRequest();//在这里创建 XMLHttpRequest对象
8. }
9. else if (window.ActiveXObject)
10. {// code for IE5 and IE6
11. xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
12. }
13. if (xmlhttp!=null)
14. {
15. xmlhttp.open("GET",url,true); //请求的方式和请求地址
16. xmlhttp.send(null);//发送请求
17. xmlhttp.onreadystatechange=state\_Change;//监听回调函数
18. }
19. else
20. {
21. alert("Your browser does not support XMLHTTP.");
22. }
23. }
24. function state\_Change() //这里是回调函数
25. {
26. if (xmlhttp.readyState==4&&xmlhttp.status==200)
27. //当满足这两个条件时表示请求成功,完成响应 4 = "loaded", 200 = OK
28. {
29. var data=xmlhttp.responseText; //拿到服务器返回的数据
30. // ...our code here...在这里进行数据返回后的操作
31. }else
32. {
33. alert("Problem retrieving XML data");
34. }
35. }

**3.原生写法中的注意点**

(1).open() 的第三个参数中使用了 "true",该参数规定请求是否异步处理，默认是异步。True 表示脚本会在 send() 方法之后继续执行，而不等待来自服务器的响应。

(2).关于readyState

(3).关于status 由服务器返回的 HTTP 状态代码，200 表示成功，而 404 表示 "Not Found" 错误。当 readyState 小于 3 的时候读取这一属性会导致一个异常。(后面会有http状态码的详细解读)

**三.JQuery中的Ajax**

JQuery对原生Ajax做了很好的封装，使用起来非常简单方便,具体的很多方法如 $.ajax，$.post， $.get， $.getJSON等能根据不同需要进行调用，写法更加简洁，但是为了兼顾各个方法在这里我以一个通用的方法 $.ajax为例做一个简单的解析,按照下面的模式写好各个参数,就能成功进行Ajax的请求了,可能在实际中使用 $.post， $.get 这两个方法使用比较多，但是**理解$.ajax 这个通用的方法能对封装原理有很好的认识。**

1. $.ajax({
2. type:   //数据的提交方式：get和post
3. url:    //请求地址
4. async:   //是否支持异步刷新，默认是true
5. data:    //需要提交的数据
6. dataType:   //服务器返回数据的类型，例如xml,String,Json等
7. success:function(data){
8. }    //请求成功后的回调函数,参数data就是服务器返回的数据
9. error:function(data){
10. }   //请求失败后的回调函数，根据需要可以不写，一般只写上面的success回调函数
11. })

**四.GET or POST？**

作为Ajax最常用的两种数据提交方式，GET和POST有着自己的特点和适用场景，**正确区分GET和POST的不同并根据实际需要进行选用在开发中十分重要，简单但是关键**！

先上一张GET 和 POST的比较图，从这张图中可以看出两者之间的差别：

**从表格中拎出关键点：** 1.传递数据的方式不同：get是直接把请求数据放在url的后面，是可见的，post的请求数据不会显示在url中，是不可见的。 2.数据长度和数据类型的差异：get有数据长度的的限制，且数据类型只允许ASCII字符，post在这两方面都没有限制。 3.安全性的差异：get不安全，post更安全。

**由此得出的两者的使用场景**：get使用较方便，适用于页面之间非敏感数据的简单传值，post使用较为安全，适用于向服务器发送密码、token等敏感数据。

**五.success和complete的区别**

JQuery封装的Ajax回调函数中，success、error、complete是最常用的三个，其中，success和error很好区别，一个是请求成功调用的，另一个是请求失败调用的，从字面上就可以理解。但是success和complete容易混淆，在这里特别做一个说明：

success:请求成功后回调函数。

complete:请求完成后回调函数 (请求成功或失败时均调用)。

注意到括号里面了吗，没错，**区别就在于complete只要请求完成，不论是成功还是失败均会调用**。也就是说如果调用了success，一定会调用complete；反过来调用了complete，不一定会调用success。(状态码404、403、301、302...都会进入complete，只要不出错就会调用)

**六.XML -> JSON**

Ajax中的是 "x" 指的就是XML。

xml:可扩展标记语言，标准通用标记语言的子集，是一种用于标记电子文件使其具有结构性的标记语言。

xml作为一种数据交互格式，广泛用在计算机领域，然而，随着json的发展，json以其明显的优势已经渐渐取代了xml成为现在数据交互格式的标准，所以在这里，想强调的是，**json现在是主流的数据交互格式**，前后端的交互标准，无论是前端提交给后台的数据，还是后台返回给前端的数据，都最好统一为json格式，各自接收到数据后再解析数据即可供后续使用。所以 "Ajax" 实际上已经发展为 "Ajaj"

**七.JSON和JSONP**

json 和 jsonp 看起来只相差了一个 “p” ，然而实际上根本不是一个东西，千万别以为是差不多的两个概念。

json：(JavaScript Object Notation, JS 对象标记) 是一种轻量级的数据交换格式。

jsonp：一种借助 <script> 元素解决主流浏览器的跨域数据访问问题的方式。

**八.Ajax跨域访问**

ajax很好，但不是万能的，**ajax的请求与访问同样会受到浏览器同源策略的限制**，不能访问不同主域中的地址。所以，为了解决这一问题，实现跨域访问，有很多种方式，上述提到的jsonp就是一种流行的方式，还有其他一些方式，我在这里就不展开说了，只是想说明ajax的使用也是有条件的，任何技术的实现都不会是没有限制的。跨域访问时一个很重要的知识点，之前专门写过一篇关于跨域访问的总结，还挺详细的，可以移步查看： javascript中实现跨域的方式总结

**九.再议HTTP状态码**

前面提到的"200"、"404"只是http状态码中常见的两个，当浏览者访问一个网页时，浏览者的浏览器会向网页所在服务器发出请求。当浏览器接收并显示网页前，此网页所在的服务器会返回一个包含HTTP状态码的信息头（server header）用以响应浏览器的请求。

需要掌握的常见http状态码大致有以下一些：

1. 101：切换协议，服务器根据客户端的请求切换协议
2. \*\*200：请求成功。一般用于GET与POST请求\*\*
3. \*\*301：永久重定向\*\*
4. \*\*302：临时重定向\*\*
5. 303：与301类似。使用GET和POST请求查看
6. \*\*304：请求资源未修改，使用缓存\*\*
7. 307：与302类似。使用GET请求重定向
8. \*\*404：客户端请求失败\*\*
9. 408：请求超时
10. \*\*500：内部服务器错误，无法完成请求\*\*
11. 505:服务器不支持请求的HTTP协议的版本，无法完成处理

**十.不可忽视的HTTP头文件**

http请求中的一个重要关注点就是请求头和响应头的内容，从这两个头文件中可以看出很多东西，当我们用发送一个ajax请求的时候，如果没有达到预期的效果，那么就需要打开浏览器的调试工具，从NetWork中找到相应的ajax请求，再通过查看请求头和响应头的信息，大体会知道这次请求的结果是怎么样的，结合响应的主体内容，可以很快找到问题。所以学会看http的头文件信息是前端开发中必须掌握的一个技能，下面就来看看具体的头文件信息。

首先随便上一张sf中的完成一个搜索结果的http请求，可以从图中的右侧清楚看到请求头和响应头的内容，包括了很多个字段信息，这些字段信息就是我们需要掌握的知识点，下面挑出其中的重点字段进行分析。

**1.请求头信息：**

1. Accept：客户端支持的数据类型
2. Accept-Charset：客户端采用的编码
3. Accept-Encoding：客户端支持的数据压缩格式
4. Accept-Language：客户端的语言环境
5. Cookie：客服端的cookie
6. Host：请求的服务器地址
7. Connection：客户端与服务连接类型
8. If-Modified-Since:上一次请求资源的缓存时间，与Last-Modified对应
9. If-None-Match：客户段缓存数据的唯一标识，与Etag对应
10. Referer:发起请求的源地址。

**2.响应头信息：**

1. content-encoding：响应数据的压缩格式。
2. content-length：响应数据的长度。
3. content-language：语言环境。
4. content-type：响应数据的类型。
5. Date:消息发送的时间
6. Age:经过的时间
7. Etag:被请求变量的实体值,用于判断请求的资源是否发生变化
8. Expires：缓存的过期时间
9. Last-Modified：在服务器端最后被修改的时间
10. server：服务器的型号

**3.两者都可能出现的消息**

Pragma：是否缓存(http1.0提出) Cache-Control:是否缓存(http1.1提出)

**4.跟缓存相关的字段**

(1) 强制缓存 expire 和 cache-control

(2) 对比缓存 Last-Modified 和 If-Modified-Since Etag 和 If-None-Match

**十一.Ajax的优缺点**

**1.优点：**

1. 页面无刷新，在页面内与服务器通信，减少用户等待时间，增强了用户体验。
2. 使用异步方式与服务器通信，响应速度更快。
3. 可以把一些原本服务器的工作转接到客户端，利用客户端闲置的能力来处理，减轻了服务器和带宽的负担，节约空间和宽带租用成本。
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。

**2.缺点：**

1. 无法进行操作的后退，即不支持浏览器的页面后退。
2. 对搜索引擎的支持比较弱。
3. 可能会影响程序中的异常处理机制。
4. 安全问题，对一些网站攻击，如csrf、xxs、sql注入等不能很好地防