什么引起了ajax不能跨域请求的问题？

ajax本身实际上是通过XMLHttpRequest对象来进行数据的交互，而浏览器出于安全考虑，不允许js代码进行跨域操作，所以会警告。

有什么完美的解决方案么？

解决方案有不少，但是只能是根据自己的实际情况来选择。

跨域的安全限制都是指浏览器端来说的，服务器端是不存在跨域安全限制的。所以针对这2种情况衍生出2类跨域解决方案，一类是服务器端做中转类似代理方式，一类是js处理浏览器端的真正跨域访问。

**具体情况有:**

1. 本域和子域的相互访问: www.aa.com和book.aa.com 用 document.domain = "aa.com";
2. 本域和其他域的相互访问: www.aa.com和www.bb.com 用 XMLHttpRequest访问代理，既服务器端代理方式
3. 本域和其他域的相互访问: www.aa.com和www.bb.com 用 JS创建动态脚本，<script>标签的src属性实现跨域访问

**解决方法：**

1. 如果想做到数据的交互，那么www.aa.com和book.aa.com必须由你来开发才可以。可以将book.aa.com用iframe添加到 www.aa.com的某个页面下,在www.aa.com和iframe里面都加上document.domain = "aa.com"，这样就可以统一域了，可以实现跨域访问。就和平时同一个域中镶嵌iframe一样，直接调用里面的JS就可以了。
2. 这种情形是最经常遇到的，也是用的最多的了。就是www.aa.com和www.bb.com你只能修改一个，也就是另外一个是别人的，人家告诉你你要取得数据就访问某某连接参数是什么样子的，最后返回数据是什么格式的。而你需要做的就是让你的服务器端充当中转代理，让服务器去别人的网站上取得数据，再返回给浏览器端。

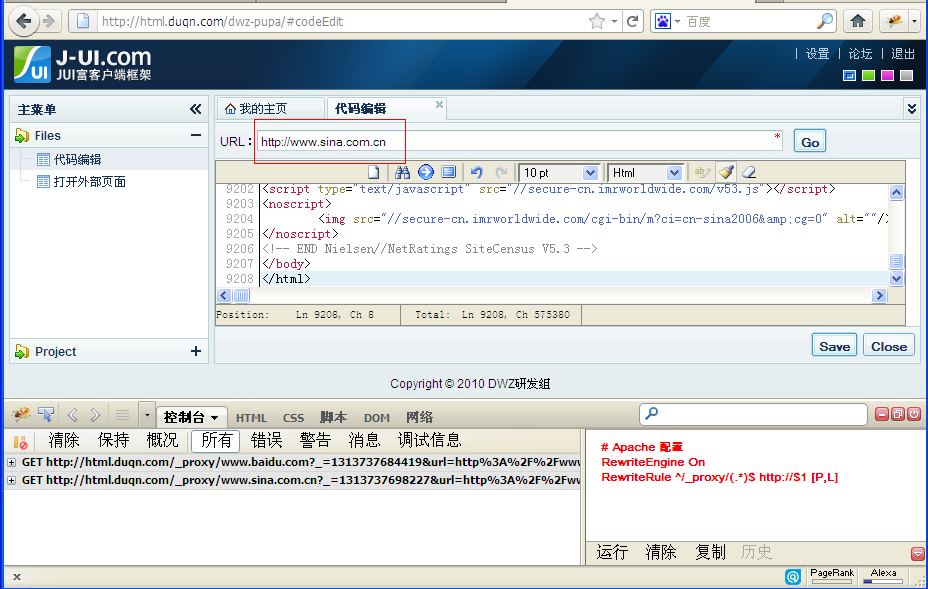
服务器端充当中转代理方式有很多可以由服务器端程序实现，也可以修改服务器配置实现，下面举例Apache重写（mod\_rewrite proxy模式）方式：  
在Apache的安装目录下的conf/httpd.conf 文件添加如下语句：

|  |
| --- |
| LoadModule proxy\_module modules/mod\_proxy.so |
| LoadModule proxy\_http\_module modules/mod\_proxy\_http.so | |

|  |  |
| --- | --- |
| LoadModule rewrite\_module modules/mod\_rewrite.so | |
| RewriteEngine On |

|  |
| --- |
| RewriteRule ^/\_proxy/(.\*)$ [http://](NULL)$1 [P,L] |

这样就可以把其他网站的url映射为本服务器的/\_proxy/目录下面, 例如可以这么访问百度 http://html.duqn.com/\_proxy/www.baidu.com

  
 这个的区别就是请求是使用<script>标签来请求的，这个要求也是两个域都是由你来开发才行。原理就是JS文件注入，在本域内的aa.com 内生成一个JS标签，它的SRC指向请求的另外一个域bb.com的某个页面b，b返回数据即可，可以直接返回JS的代码。因为script的src属性是可以跨域的。具体看代码，这个也比较简单。

aa.com/a.jsp

|  |  |
| --- | --- |
| <script type="text/javascript"> | |
| function myTest(data) { |

|  |  |
| --- | --- |
| alert(data); | |
| } |

|  |
| --- |
| </script> |
| <script type="text/javascript" src="<http://www.bb.com/index>!getData.action?jsoncallback=myTest"></script> | |

bb.com/b.jsp页面代码如下：

|  |
| --- |
| $(param.jsoncallback)({"name": "Zhang Huihua", "QQ": "350863780"}) |

b.jsp页面通过 $(param.jsoncallback)得到浏览器端随后要回调的js function name:myTest

实际上客户端接收到的response如下：myTest({"name": "Zhang Huihua", "QQ": "350863780"})

**jQuery浏览器端跨域访问**

目前jQuery $.ajax()支持get方式的跨域，这其实是采用jsonp的方式来完成的。其原理就是上面第三种方式，<script>标签的src属性实现跨域访问

真实案例代码如下:

|  |
| --- |
| $.ajax({ |
| url: http://跨域的dns/index!searchJSONResult.action, | |

|  |
| --- |
| type: "GET", |
| dataType: 'jsonp', | |

|  |  |
| --- | --- |
| data: {name:”ZhangHuihua”}, | |
| timeout: 5000, |

|  |  |
| --- | --- |
| success: function (json) {//客户端jquery预先定义好的callback函数,成功获取跨域服务器上的json数据后,会动态执行这个callback函数 | |
| alert(json); |

|  |
| --- |
| }, |
| error: function (xhr, ajaxOptions, thrownError){ | |

|  |  |
| --- | --- |
| alert("Http status: " + xhr.status + " " + xhr.statusText + "\najaxOptions: " + ajaxOptions + "\nthrownError:"+thrownError + "\n" +xhr.responseText); | |
| } |

|  |
| --- |
|  |
| }); | |

注意:

|  |  |
| --- | --- |
| $.getJSON("[http://](NULL)跨域的dns/index!searchJSONResult.action?name1="+value1+"&jsoncallback=?", function(json){ | |
| // 执行代码 |

|  |
| --- |
| }); |

这种方式其实是上例$.ajax({..}) api的一种高级封装,有些$.ajax api底层的参数就被封装而不可见了.

这样,jquery就会拼装成如下的url get请求

http:// 跨域的 dns/index!searchJSONResult.action?&jsoncallback=jsonp1236827957501&\_=1236828192549&name=ZhangHuihua

在响应端(http://跨域的dns/index!searchJSONResult.action),

通过 jsoncallback = request.getParameter("jsoncallback") 得到jquery端随后要回调的js function name:jsonp1236827957501

然后 response的内容为一个Script Tags:"jsonp1236827957501("+按请求参数生成的json数组+")";

jquery就会通过回调方法动态加载调用这个js tag:jsonp1236827957501(json数组);

这样就达到了跨域数据交换的目的.

jsonp的最基本的原理是:动态添加一个<script>标签，而script标签的src属性是没有跨域的限制的。这样说来,这种跨域方式其实与ajax XmlHttpRequest协议无关了.

这样其实"jQuery AJAX跨域问题"就成了个伪命题了,jquery $.ajax方法名有误导人之嫌.

如果设为dataType: 'jsonp', 这个$.ajax方法就和ajax XmlHttpRequest没什么关系了,取而代之的则是JSONP协议.

JSONP是一个非官方的协议，它允许在服务器端集成Script tags返回至客户端，通过javascript callback的形式实现跨域访问

JSONP即JSON with Padding。由于同源策略的限制，XmlHttpRequest只允许请求当前源（域名、协议、端口）的资源。如果要进行跨域请求，

我们可以通过使用html的script标记来进行跨域请求，并在响应中返回要执行的script代码，其中可以直接使用JSON传递javascript对象。

这种跨域的通讯方式称为JSONP。

jsonCallback 函数jsonp1236827957501(....): 是浏览器客户端注册的，获取跨域服务器上的json数据后，回调的函数

Jsonp原理：

首先在客户端注册一个callback (如:'jsoncallback'), 然后把callback的名字(如:jsonp1236827957501)传给服务器。

此时，服务器先生成 json 数据。

然后以 javascript 语法的方式，生成一个function , function 名字就是传递上来的参数 'jsoncallback'的值 jsonp1236827957501 .

最后将 json 数据直接以入参的方式，放置到 function 中，这样就生成了一段 js 语法的文档，返回给客户端。

客户端浏览器，解析script标签，并执行返回的 javascript 文档，此时javascript文档数据,作为参数，

传入到了客户端预先定义好的 callback 函数(如上例中jquery $.ajax()方法封装的的success: function (json))里.（动态执行回调函数）

可以说jsonp的方式原理上和<script src="http://跨域/...xx.js"></script>是一致的(qq空间就是大量采用这种方式来实现跨域数据交换的) .JSONP是一种脚本注入(Script Injection)行为,所以也有一定的安全隐患.

注意,jquey是不支持post方式跨域的.

虽然采用post +动态生成iframe是可以达到post跨域的目的,但这样做是一个比较极端的方式,不建议采用.

也可以说get方式的跨域是合法的,post方式从安全角度上,被认为是不合法的, 万不得已还是不要剑走偏锋..

浏览器端跨域访问的需求看来也引起w3c的注意了,看资料说html5 WebSocket标准支持跨域的数据交换,应该也是一个将来可选的跨域数据交换的解决方案.