

关于跨域和jsonp的一些理解(新手向)

本文内容浅显，适合人群：

- 1.不懂什么是跨域
- 2.不想自己弄两个不同域名进行跨域测试的童鞋

什么是跨域

我的理解是，当用户对不同协议或不同端口或不同域名的资源进行访问时，就是跨域。

为什么会造成跨域

罪魁祸首：同源策略

同源定义：即同一域，即相同协议&相同端口&相同域名&相同子域名

同源策略规定：XHR对象只能访问与包含它的页面位于同一域中的资源，有利于预防一些恶意行为。

怎么解决跨域

解决办法有很多，CORS、iframe、h5新特性postMessage等，而比较简单的方法就是今天着重介绍的jsonp。

解决依据：尽管不能访问非本域的动态资源，但是类似js文件、样式、图片等静态资源是可以访问的！就是通过这个“漏洞”来解决跨域问题，用 `<script>` 标签中的src来写入跨域数据的url，这样就能绕过同源策略了。

简单介绍jsonp

JSONP，JSON with Padding的简写，这个全称对jsonp的理解还是有一定的帮助的。填充式JSON或者说是参数式JSON。JSONP的语法和JSON很像，简单来说就是在JSON外部用一个函数包裹着。JSONP基本语法如下：

```
callback({ "name": "kwan", "msg": "获取成功" });
```

JSONP两部分组成：回调函数和里面的数据。回调函数是当响应到来时，应该在页面中调用的函数，一般是在发送过去的请求中指定。

JSONP原理：

刚才的解决依据可知，JSONP原理就是动态插入带有跨域url的 `<script>` 标签，然后调用回调函数，把我们需要的json数据作为参数传入，通过一些逻辑把数据显示在页面上。

原理看千遍，一写又不会

我想说这个JSONP我看了很多次，参考过不同的文章，原理都会背了，结果自己写一个出来，傻眼。。

网上有很多文章，良心一点的会配有些代码解释，然而对小白来说，要自己down个WAMP什么的服务器集成软件或者自己搭个服务器，再弄两个不同源的html才能进行模拟跨域，这真的。。。好麻烦。。

终于，看到了一篇文章，一个通过调用api接口来模拟跨域请求数据的DEMO，实在感谢啊！这才是我所要的好吗！（链接在本文最后）

JSONP跨域小实践

你再怎么懂原理，再怎么会JSONP基本语法，不真正执行一下你其实还是懵逼。

当然，我认为我还是需要模拟不同源的跨域请求才对真正弄懂跨域有一定好处。

下面展示的代码源于那篇文章，需求是输入歌名，点击搜索后，跨域请求API接口，返回数据后，显示专辑名在页面。

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>JSONP</title>
6  </head>
7  <body>
8      <input type="text" id="song" name="">
9      <input type="button" id="song_search" value="歌曲搜索" name="">
10     <br />
11     <div style="width:200px;height:230px;background:pink" id="song_list"></div>
12     <script type="text/javascript" src="http://code.jquery.com/jquery-
13 2.1.1.min.js"></script>
14     <script type="text/javascript">
15         var searchJsonCallback=function(data){
16             //遍历查询结果
17             var alb_html='';
18             for(var i in data.list){
19                 alb_html+= '<span>专辑: </span><div
20 style="color:black">'+data.list[0].albumname+'</div>';
21             }
22             $("#song_list").html(alb_html);
23         };
24         $("#song_search").on("click",function(){
25             var keyword=$("#song").val();
26             if(keyword==undefined||keyword==""){
27                 alert("歌曲搜索不能为空");
28                 return false;
29             }else{
30                 var url = "http://cgi.music.soso.com/fcgi-
31 bin/fcg_search_xmldata.q?source=10&w="+keyword+"&perpage=1&ie=utf-8";
```

```

29         // 创建script标签, 设置其属性
30         var script = document.createElement('script');
31         script.setAttribute('src', url);
32         // 把script标签加入head, 此时调用开始
33         document.getElementsByTagName('head')[0].appendChild(script);
34     }
35
36     });
37
38     </script>
39 </body>
40 </html>

```

下面是我对以上代码的一些文字解释：

1) 点击按钮后动态插入跨域数据，然后由于此接口用searchJsonCallback({})来封装json格式数据，因此可看作是调用一个函数，同时把json数据作为参数传入

2) 所以当动态插入script标签后，写好了的searchJsonCallback()函数将被调用，参数data就是json数据，然后通过遍历渲染到DOM上，完成整个跨域获取数据流程

总结步骤：触发click事件-动态插入带有API接口的script标签-根据回调函数名调用函数-遍历数据-渲染到页面

当JSONP遇上jQuery ajax

注意了，千万不要认为用了jQuery ajax，就是通过它来实现跨域请求，其实只是因为它很好地封装了JSONP而已

用jQuery ajax原理和上面的是一样的，只不过我们不需要手动的插入script标签以及定义回调函数。jquery会自动生成一个全局函数来替换callback=?中的问号，之后获取到数据后又会自动销毁，实际上就是起一个临时代理函数的作用。

于是对上面的代码进行修改，并且把请求数据改为每次显示5条：

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <title>JSONP2</title>
6  </head>
7  <body>
8  <input type="text" id="song" name="">
9  <input type="button" id="song_search" value="歌曲搜索" name="">
10 <br />
11 <div style="width:200px;height:230px;background:pink" id="song_list"></div>
12 <script type="text/javascript" src="http://code.jquery.com/jquery-
13 2.1.1.min.js"></script>
14 <script type="text/javascript">

```

```

15     $("#song_search").on("click",function(){
16         var keyword=$("#song").val();
17         if(keyword==undefined||keyword==""){
18             alert("歌曲搜索不能为空");
19             return false;
20         }else{
21             $.ajax({
22                 url: 'http://cgi.music.soso.com/fcgi-bin/fcg_search_xmldata.q?
source=10&w='+keyword+'&perpage=5&ie=utf-8',
23                 type: 'GET',
24                 dataType: 'jsonp',
25                 jsonp: 'callback',
26                 jsonpCallback: 'searchJsonCallback',
27                 success:function(data){
28                     //遍历查询结果
29                     var alb_html=''; //创建一个变量用于DOM拼接
30                     for(var i in data.list){
31                         alb_html+=''<p
32                         style="color:black">'+data.list[i].albumname+'</p>';
33                         console.log(data.list[i].singername);
34                     }
35                     var name_html='<span>专辑: </span>';
36                     var sum_html=name_html+alb_html;
37                     $("#song_list").html(sum_html);
38                 }
39             });
40         }
41     });
42 </script>
43 </body>
44 </html>

```

关于jsonp、jsonpCallback两个属性，部分解释一直没看懂，如下：

jsonp: "callback",

//传递给请求处理程序或页面的，用以获得jsonp回调函数名的参数名(一般默认为:callback)

jsonpCallback:"handler",

//自定义的(?)jsonp回调函数名称，默认为jQuery自动生成的随机函数名，也可以写"?"，jQuery会自动为你处理数据

不懂直接看Network。其实就是拼接到url时想要显示的内容。

如上面情况，url则会变成"http://...&perpage=5&ie=utf-8&callback=searchJsonCallback&..."

所以简单来说，

jsonp就是相当于一个名字，一个参数名

jsonpCallback就是回调函数名，用来包裹JSON数据的

然后将jsonpCallback定义的函数赋值给jsonp定义的参数上，

最后拼接在url末尾完成参数传递。

注意：jsonp可以随便写，不写也行，反正默认是callback

但jsonpCallback必须是对应其返回数据的函数名，此API接口返回的数据外围用searchJsonCallback()包裹数据，因此只能是这个

（有些文章说这个也是可以自定义的，其实是可以自定义的，但是因为这个API接口固定了函数名,所以只能写成这个）

参考资料：

上面提到的原文链接：（可惜没有CORS举例 😞）

<http://www.cnblogs.com/st-les...>

另一篇帮助理解的文章：

<http://blog.csdn.net/u0139455...>