

```
$.ajax({
    url:"<%=saveMobileColumn%>siteId="+vsiteId+"&columnIds="+couolumnIds,
    type:"post",
    dataType:"jsonp",
    jsonp:"callback",
    success:function(result) {
        var data = eval(JSON.stringify(result.data));
        var das =eval(data);
        if(das[0].state == "success"){

            layer.alert(' 设置成功! ', {closeBtn: 0,title: '提示'});

        }else{
            layer.alert(das[0].state, {closeBtn: 0,title: '提示'});
        }
    }
});
```

轻松搞定JSONP跨域请求

<http://blog.csdn.net/u014607184/article/details/52027879>

一、同源策略

要理解跨域，先要了解一下“同源策略”。所谓同源是指，域名，协议，端口相同。所谓“同源策略“，简单的说就是基于安全考虑，当前域不能访问其他域的东西。

一些常见的是否同源示例可参照下表：

URL	是否同源	
http://www.a.com https://www.a.com	不同源	协议
http://www.a.com http://www.b.com	不同源	域名
http://www.a.com : 8080 http://www.a.com : 1000	不同源	端口
http://www.a.com/a.js http://www.a.com/b.js	同源	协议、域名
http://www.a.com/test1/a.js http://www.a.com/test2/b.js	同源	同一域名

这里写图片描述

在同源策略下，在某个服务器下的页面是无法获取到该服务器以外的数据的。例如我们在自己的网站通过ajax去获取豆瓣上

https://developers.douban.com/wiki/?title=api_v2提供的接口数据。这里我们以搜索图书为例，参数链接为：

<https://api.douban.com/v2/book/search?q=javascript&count=1>，该链接中数据为JSON格式，如下：

```
{
  "count":1,"start":0,"total":1275,"books":[{"rating":
    {"max":10,"numRaters":77,"average":9.2,"min":0},"subtitle":"The Good Parts","author":["克洛克福
    特"],"pubdate":"2009-1","tags":[{"count":94,"name":"javascript","title":"javascript"},
    {"count":39,"name":"编程","title":"编程"}, {"count":29,"name":"web","title":"web"},
    {"count":27,"name":"JavaScript","title":"JavaScript"}, {"count":22,"name":"技术","title":"技术"},
    {"count":18,"name":"programming","title":"programming"}, {"count":15,"name":"js","title":"js"},
    {"count":15,"name":"程序设计","title":"程序设计"}],
    "origin_title":"","image":"https://img3.doubanio.com/mpic/s3400022.jpg","binding":"平
    装","translator":[],"catalog":"","pages":"153","images":
    {"small":"https://img3.doubanio.com/spic/s3400022.jpg","large":"https://img3.doubanio.com/lpic/s34000
    22.jpg","medium":"https://img3.doubanio.com/mpic/s3400022.jpg"},"alt":"https://book.douban.com/subject/3332698/","id":"3332698","publisher":"东南大学出版社","isbn10":"7564114479","isbn13":"9787564114473","title":"JavaScript","url":"https://api.douban.com/v2/book/3332698","alt_title":"","author_intro":"","summary":"《JavaScript:The Good Parts(影印版)》一书中,Crockford深度分析了一堆好的意图和盲目的错误,为你提供了所有JavaScript的地道优良部分的细节,包括:·语法·继承·方法;·对象·数组·风格;·函数·正则表达式·美丽的特性大多数编程语言包含优良和拙劣的部件,但对JavaScript而言后者的比重较大,因为它在匆忙中开发和发布,还不能够得到精炼。这本权威的书剔除了大多数可怕的JavaScript特性,展现了JavaScript的另一部分,这一部分比JavaScript语言作为一个整体更加稳定、更具有可读性以及可维护性——可以用这个部分创建真正可展的合高效的代码。\\n作者Douglas Crockford(他被很多开发社区认为是JavaScript专家)提出了足够多的好想法,让JavaScript成为一个杰出的面向对象编程语言。不幸的是,这些好想法(比如函数、弱类型、动态对象和表达能力很强的对象文字注释)被掺杂了些坏想法(比如基于全局变量的编程模型)。\\n当Java Applet陨落的时候,JavaScript成为了Web编程的缺省语言,但它的流行程度跟它作为一个编程语言的质量完全没有关系。","price":"28.00元"}]}
```

这里写图片描述

我通过如下代码去访问该数据:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>ajax</title>
</head>
<body>
  <div id="mydiv">
    <button id="btn">点击</button>
  </div>
</body>
<script type="text/javascript">
  window.onload = function() {

    var oBtn = document.getElementById('btn');

    oBtn.onclick = function() {

      var xhr = new XMLHttpRequest();


      xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
          alert( xhr.responseText );
        }
      };

      xhr.open('get', 'https://api.douban.com/v2/book/search?q=javascript&count=1', true);
      xhr.send();
    };
  };
</script>
</html>
```

上述程序运行时会报错:

XMLHttpRequest cannot load https://api.douban.com/v2/book/search?q=javascript&count=1. No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://localhost' is therefore not allowed access. ajax.html:1

这里写图片描述

但的src (获取图片), <link>的href (获取css), <script>的src (获取javascript) 这三个都不符合同源策略, 它们可以跨域获取数据。这里要介绍的JSONP就是利用<script>的src来实现跨域获取数据的。

二、JSONP

JSONP 是 JSON with padding（填充式 JSON 或参数式 JSON）的简写。

JSONP实现跨域请求的原理简单的说，就是动态创建<script>标签，然后利用<script>的src 不受同源策略约束来跨域获取数据。

JSONP 由两部分组成：回调函数和数据。回调函数是当响应到来时应该在页面中调用的函数。回调函数的名字一般是在请求中指定的。而数据就是传入回调函数中的 JSON 数据。

动态创建<script>标签，设置其src，回调函数在src中设置：

```
var script = document.createElement("script");
script.src = "https://api.douban.com/v2/book/search?q=javascript&count=1&callback=handleResponse";
document.body.insertBefore(script, document.body.firstChild);
```

在页面中，返回的JSON作为参数传入回调函数中，我们通过回调函数来来操作数据。

```
function handleResponse(response) {
    // 对response数据进行操作代码
}
```

了解了JSONP的基本使用方法，我们在实现上面通过ajax调用豆瓣接口的需求，实现代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>JSONP实现跨域2</title>
</head>
<body>
    <div id="mydiv">
        <button id="btn">点击</button>
    </div>
</body>
<script type="text/javascript">
    function handleResponse(response) {
        console.log(response);
    }
</script>
<script type="text/javascript">
    window.onload = function() {

        var oBtn = document.getElementById('btn');

        oBtn.onclick = function() {

            var script = document.createElement("script");
            script.src = "https://api.douban.com/v2/book/search?q=javascript&count=1&callback=handleResponse";
            document.body.insertBefore(script, document.body.firstChild);

        };
    };
</script>
</html>
```

在控制台，我们可以查看到返回的response数据格式为JSON对象格式的，具体需要取出哪些参数，可以根据自己的需要：

```

▼ Object {count: 1, start: 0, total: 1275, books: Array[1]} ⓘ
  ▼ books: Array[1]
    ▼ 0: Object
      alt: "https://book.douban.com/subject/3332698/"
      alt_title: ""
      ▶ author: Array[1]
        author_intro: ""
        binding: "平装"
        catalog: ""
        id: "3332698"
        image: "https://img3.doubanio.com/mpic/s3400022.jpg"
      ▶ images: Object
        isbn10: "7564114479"
        isbn13: "9787564114473"
        origin_title: ""
        pages: "153"
        price: "28.00元"
        pubdate: "2009-1"
        publisher: "东南大学出版社"
      ▶ rating: Object
        subtitle: "The Good Parts"
        summary: "《JavaScript: The Good Parts(影印版)》一书中，Crockford深度分析了一堆好的意图和盲目的错误，为你提供了所有JavaScript的地址"
      ▶ tags: Array[8]
        title: "JavaScript"
      ▶ translator: Array[0]
        url: "https://api.douban.com/v2/book/3332698"
      ▶ __proto__: Object
      length: 1
      ▶ __proto__: Array[0]
    count: 1
    start: 0
    total: 1275
    ▶ __proto__: Object

```

这里写图片描述

JSONP目前还是比较流行的跨域方式，虽然JSONP使用起来方便，但是也存在一些问题：

首先，JSONP 是从其他域中加载代码执行。如果其他域不安全，很可能在响应中夹带一些恶意代码，而此时除了完全放弃JSONP 调用之外，没有办法追究。因此在使用不是你自己运维的 Web 服务时，一定得保证它安全可靠。

其次，要确定JSONP 请求是否失败并不容易。虽然HTML5 给<script>元素新增了一个 onerror事件处理程序，但目前还没有得到任何浏览器支持。为此，开发人员不得不使用计时器检测指定时间内是否接收到了响应。

二、jQuery封装JSONP

对于经常用jQuery的开发者来说，能注意到jQuery封装的\$.ajax中有一个dataType属性，如果将该属性设置成dataType:"jsonp"，就能实现JSONP跨域了。需要了解的一点是，虽然jQuery将JSONP封装在\$.ajax中，但是其本质与\$.ajax不一样。

通过jQuery的\$.ajax实现跨域的代码参考如下：

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>jQuery实现JSONP</title>
</head>
<body>
  <div id="mydiv">
    <button id="btn">点击</button>
  </div>
</body>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.min.js"></script>
<script type="text/javascript">
  $(function() {
    $("#btn").click(function() {

      $.ajax({
        async : true,
        url : "https://api.douban.com/v2/book/search",
        type : "GET",
        dataType : "jsonp", // 返回的数据类型，设置为JSONP方式
        jsonp : 'callback', //指定一个查询参数名称来覆盖默认的 jsonp 回调参数名 callback
        jsonpCallback: 'handleResponse', //设置回调函数名
        data : {

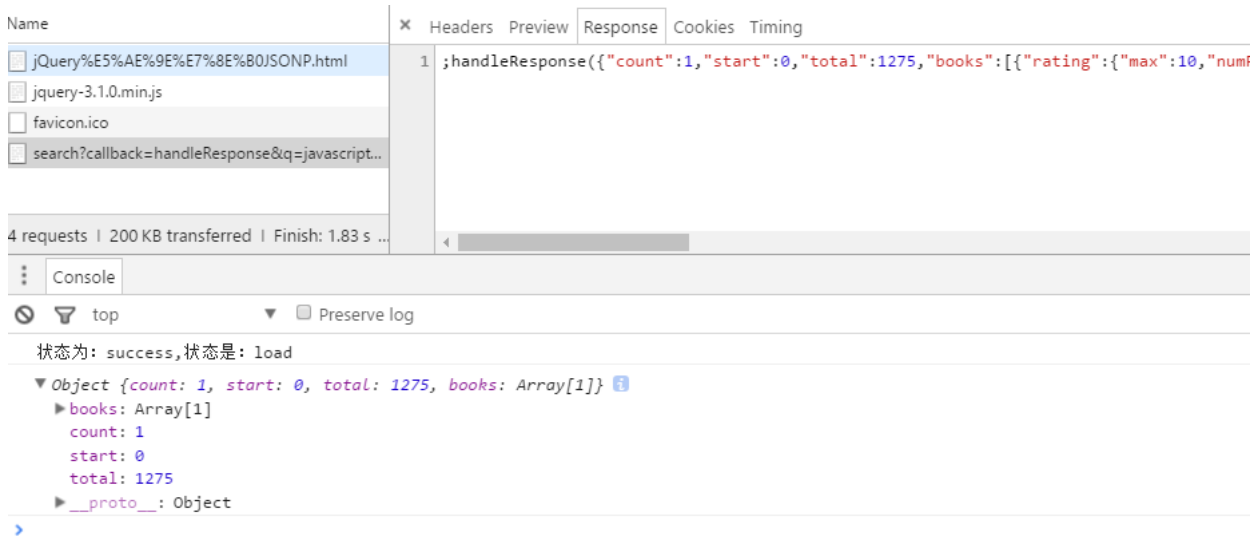
```

```

        q : "javascript",
        count : 1
    },
    success: function(response, status, xhr){
        console.log(' 状态为: ' + status + ', 状态是: ' + xhr.statusText);
        console.log(response);
    }
});
});
</script>
</html>

```

最后的结果与JavaScript通过动态添加<script>标签得到的结果是一样的。



这里写图片描述

通过\$.getJSON()

利用getJSON来实现，只要在地址中加上callback=?参数即可，参考代码如下：

```

$.getJSON("https://api.douban.com/v2/book/search?q=javascript&count=1&callback=?", function(data) {
    console.log(data);
});

```

这样也能实现跨域的功能。