

# JSON和JSONP的区别，以及使用方法

作者: [oldcaptain](#)

原文链接: <https://ld246.com/article/1378820709143>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## <h2> (一) 场景</h2>

<p>在拉京东商城选择的基础数据时候，遇到被服务器拒绝的情况，也就是ajax跨域问题</p>

## <h2> (二) json和jsonp</h2>

<p>说的直白一点，在我们做ajax异步的一些功能的时候，一定会或多或少的遇到两个问题，（1、数据的交换。2、跨域问题）</p>

<p><span>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。我们在用其关于src的标签的时候，都可以跨域。所以我们只要动态构造的src标签同样可以做到跨域。</span></p>

```
<pre class="brush: js"><script type="text/javascript">&gt;
function <span style="color: #ff0000;">jsonpCallback</span>(result)
```

```
    {
        alert(result.msg);
    }
</script>
```

```
<script type="text/javascript" src="http://crossdomain.com/jsonServerResponse?jsonp=">jsonpCallback</script>&gt;</script>
</pre>
```

<p><span>从红色的地方可以看出，两个名字要一直，不过一般采用callback然后后面加一个“?”此时jquery会自动生成一个函数名，这样可以做到发送很多请求的时候，并不会互相影响。</span></p>

<p><span><span>同源策略：<span>即JavaScript只能访问与包含它的文档在同一域下的内容。</span></span></span></p>

<p><span><span><span>jsonp可以跨越同源策略，当我们使用了jsonp，将会是另外一种协议信了。</span></span></span></p>

<p>JSONP的优点是：它不像XMLHttpRequest对象实现的Ajax请求那样受到同源策略的限制；它兼容性更好，在更加古老的浏览器中都可以运行，不需要XMLHttpRequest或ActiveX的支持；并且请求完毕后可以通过调用callback的方式回传结果。</p>

<p>JSONP的缺点则是：它只支持GET请求而不支持POST等其它类型的HTTP请求；它只支持跨域HTTP请求这种情况，不能解决不同域的两个页面之间如何进行JavaScript调用的问题。</p>

## <h2><span><span><span> (三) 使用方法</span></span></span></h2>

```
<pre class="brush: js">$.getJSON("http://跨域的dns/document!searchJSONResult.action?name1="&value1+"&jsonpcallback=?&quot;,,
```

```
    function(json){
        if(json.属性名==值){
            // 执行代码
        }
    }
});
```

```
</pre>
<pre class="brush: js">$.ajax({
    async:false,
    url: http://跨域的dns/document!searchJSONResult.action,
    type: "GET",
    dataType: 'jsonp',
</pre>
```

<p><span><span><span>通常情况一般是这两种使用方法，在跨域问题上，非常简单。</span></span></span></p>

<p><span><span><span>在这里附一个json的处理方法</span></span></span></p>

```
<pre class="brush: java">public static Object[] getDTOArray(String jsonString, Class<?&gt;
lazz){
    net.sf.json.JSONArray array = net.sf.json.JSONArray.fromObject(jsonString);
    Object[] obj = new Object[array.size()];
    for(int i = 0; i < array.size(); i++){
        net.sf.json.JSONObject jsonObject = array.getJSONObject(i);
```

```
obj[i] = net.sf.json.JSONObject.toBean(jsonObject, clazz);
}
return obj;
}</pre>
<p><span><span><span>&nbsp;</span></span></span></p>
```