



链滴

前后端分离实践——Jsonp 数据交互

作者: [liumapp](#)

原文链接: <https://ld246.com/article/1510150517412>

来源网站: [链滴](#)

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

<blockquote>
<p>笔者这里所说的前后端分离，是指后端通过 SpringBoot 解决业务逻辑，视图及交互通过 Vue 解决，两者之间的数据交互则通过 Jsonp 实现。这种做法完全摒弃了传统 MVC 设计模式里面的模板引擎，为何这样做？嗯...通俗点说，当你体验过 Vue 有多美，就能明白模板引擎有多丑了...</p>
</blockquote>
<h3 id="项目源代码">项目源代码</h3>
<p>个人比较喜欢在讲一个东西之前，先把自己准备的 Demo 代码放出来，因为对于牛逼的看官而，有代码就足够了。</p>
<p>Github 测试项目: simple-jsonp </p>
<p>以及 Vue 插件项目: simple-jsonp-plugin </p>
<p>simple-jsonp 是主程序，simple-jsonp-plugin 是一个 Vue 的插件，该插件封装了对 Vuex store 的使用和利用 JQuery 下的 jsonp 来获取后端数据，并写入 state 中。</p>
<p>程序不难，但是涉及到的知识面稍微有点点范，可能会要求看官有一定的全栈知识才能完全理解</p>
<p>至于 Jsonp 是什么，看官可以参考我的另一篇博文: Jsonp 原理 </p>
<h3 id="前言">前言</h3>
<p>因为在本本地同时部署了前后端的环境，后端监听 9080 端口，前端监听 8080 端口，所以在前后数据 ajax 交互上面，就涉及到了跨域的问题。</p>
<p>请各位注意，跨域并不仅仅是说域名不一致会产生，端口不一致也是会发生这个问题的。</p>
<p>在解决跨域这个问题的前提下，首先我们需要准备一个环境，这个环境，也就是说，你的前端是么，后端是什么，然后再来进行数据交互，把跨域问题产生了，再来解决它。</p>
<p>在解决问题的具体方式上，我没有直接写在前端的代码里，而是另外新建了一个 Vue 插件，在这个插件中，去执行 Jsonp 的跨域请求数据。</p>
<p>现在先说一下前后端的具体环境。</p>
<h3 id="具体环境">具体环境</h3>
<p>后端基于 Maven 管理依赖，SpringBoot 驱动。前端则通过 Vue 全家桶套餐来实现。具体列表下: </p>
<h4 id="后端">后端</h4>

<p>Maven</p>

<p>SpringBoot</p>

<h4 id="前端">前端</h4>

<p>Vue2.0</p>

<p>Vue-router</p>

<p>Vuex</p>

<p>Vue-devtools</p>


```

</li>
<li>
<p>加一个简单的 Vue 插件</p>
</li>
</ul>
<p>接下来开始我那不太正式的分析</p>
<h3 id="分析">分析</h3>
<h5 id="后端">后端</h5>
<p>后端不多说了，直接启动，然后等待前端的数据请求即可。</p>
<p>但是要注意一点：</p>
<p>后端在返回数据的时候，一定是：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">callback({status:1, content: 'hello jsonp' , message: 'hello world'})
</span></span></code></pre>
<p>这种，在一个方法名称内包含一个 Json 对象的格式。</p>
<p>同时建议使用 FastJson 或者其他 Json 转换的工具完成 json 的转换。</p>
<p>因为我们需要考虑，对象转 json、数组转 json 等等场景。</p>
<h5 id="前端">前端</h5>
<p>在我写的 Vue 插件中，直接跟 Jsonp 相关的代码就是以下这段：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">opt.$.ajax({
</span></span><span class="highlight-line"><span class="highlight-cl">  type:"get",
</span></span><span class="highlight-line"><span class="highlight-cl">  async:false,
</span></span><span class="highlight-line"><span class="highlight-cl">  url:opt.url,
</span></span><span class="highlight-line"><span class="highlight-cl">  dataType:'jsonp'

</span></span><span class="highlight-line"><span class="highlight-cl">  jsonpCallback:'
allback',
</span></span><span class="highlight-line"><span class="highlight-cl">  success: function (
json) {
</span></span><span class="highlight-line"><span class="highlight-cl">    console.log(js
n);
</span></span><span class="highlight-line"><span class="highlight-cl">    opt.store.co
mit('setStatus' , {
</span></span><span class="highlight-line"><span class="highlight-cl">      status: json.s
atus
</span></span><span class="highlight-line"><span class="highlight-cl">    });
</span></span><span class="highlight-line"><span class="highlight-cl">    opt.store.co
mit('setContent' , {
</span></span><span class="highlight-line"><span class="highlight-cl">      content: json
content
</span></span><span class="highlight-line"><span class="highlight-cl">    });
</span></span><span class="highlight-line"><span class="highlight-cl">    opt.store.co
mit('setMsg' , {
</span></span><span class="highlight-line"><span class="highlight-cl">      msg: json.me
sage
</span></span><span class="highlight-line"><span class="highlight-cl">    });
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  error: function (
rror) {

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">    console.log('s
mething wrong')
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl">  })
</span></span></code></pre>
<p>其中，opt.store 的内容如下所示： </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">import Vue from 'vue'
</span></span><span class="highlight-line"><span class="highlight-cl">import Vuex from
vuex'
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Vue.use(Vuex)
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">const state = {
</span></span><span class="highlight-line"><span class="highlight-cl">  status: null,
</span></span><span class="highlight-line"><span class="highlight-cl">  content: null,
</span></span><span class="highlight-line"><span class="highlight-cl">  msg: null
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">const mutations =
{
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">  setStatus (state,
ayload) {
</span></span><span class="highlight-line"><span class="highlight-cl">    state.status = p
ayload.status
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">  setContent (state
payload) {
</span></span><span class="highlight-line"><span class="highlight-cl">    state.content =
payload.content
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">  setMsg (state, pa
load) {
</span></span><span class="highlight-line"><span class="highlight-cl">    state.msg = pay
oad.msg
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">export default ne
Vuex.Store({
</span></span><span class="highlight-line"><span class="highlight-cl">  state,
</span></span><span class="highlight-line"><span class="highlight-cl">  mutations
</span></span><span class="highlight-line"><span class="highlight-cl">})
</span></span></code></pre>
<p>它的作用很简单，就是用来将从后端获取到的数据，写入 Vuex 的 state 中，供其他的 compon
nt 调用。 </p>
<p>opt.$ 就是指从 Vue 中获取到的 JQuery，我们利用 jquery 对 Jsonp 的完美支持，来实现前端
后端 jsonp 数据请求及回调方法的使用。 </p>
<p>再简单点来说，把 dataType 设置为 jsonp，把 jsonpCallback 设置的值跟后端返回的方法名保
一致，即可以正常使用，success 方法中传入的 json 对象，便是后端在方法中加入的 json 对象，我
直接拿过来用即可。 </p>

```

<p>写到这里，我们便完成了前后端 jsonp 的数据交互。 </p>