前后端数据交互方法

**前后端数据交互方法**

在此介绍几种常用的前后端数据交互方法，并给出使用建议。以提高前后端协同开发的效率。  
  
此文章适合前后端协同开发经验不足的新手阅读。

**目录:**

1. [HTML赋值](https://github.com/nimojs/blog/issues/13#hash_htmlvalue1)
2. [JS赋值](https://github.com/nimojs/blog/issues/13#hash_jsvar2)
3. [script填充JSON](https://github.com/nimojs/blog/issues/13#hash_json3)
4. [AJAX获取JSON](https://github.com/nimojs/blog/issues/13#hash_ajaxjson4)
5. [WebSocket实时传输数据](https://github.com/nimojs/blog/issues/13#hash_websocket5)
6. [总结](https://github.com/nimojs/blog/issues/13#hash_summary6)

**HTML赋值**

输出到 Element 的 value 或 data-name

<input type="hidden" value="<?php echo $user\_avatar;?>" />

<div data-value="<?php echo $user\_avatar;?>"></div>

渲染结果

<input type="hidden" value="https://avatars1.githubusercontent.com/u/3949015?v=3&s=40" />

<div data-avatar="https://avatars1.githubusercontent.com/u/3949015?v=3&s=40"></div>

使用 JS 获取

$('input').val();

// http://jquery.bootcss.com/jQuery.data/

$('div').data('avatar');

**优点：**  
  
不占用全局变量，由 JS 自由获取。

**使用建议：**

适合传递简单数据，也非常适合多个简单数据与 Element 绑定关系。

<ul>

<li>nimojs <span data-userid="1" >删除</span></li>

<li>nimo22 <span data-userid="2" >删除</span></li>

<li>nimo33 <span data-userid="3" >删除</span></li>

<li>nimo44 <span data-userid="4" >删除</span></li>

<li>nimo55 <span data-userid="5" >删除</span></li>

</ul>

<script>

$('span').on('click',function(){

$.post('/ajax/remove/',$(this).data('userid'),function(data){

// ...

})

})

</script>

**JS赋值**

将数据填充到 <script> 的 JavaScript 变量声明中。

<script>

var user\_avatar = "<?php echo $user\_avatar;?>";

// 渲染结果

// var user\_avatar = "https://avatars1.githubusercontent.com/u/3949015?v=3&s=40";

</script>

或使用 Smarty 后端模板引擎：

<script>

var user\_avatar = "{$user\_avatar}";

</script>

**优点：**  
传递数据非常方便。前端直接调用 user\_avatar 变量使用数据。

**缺点：**  
1. 为了传递一个字符串数据占用了全局变量 user\_avatar，当有很多数据需要传输时则会占用很多全局变量。  
2. 如果返回数据存在换行将会导致JS报错

// 渲染结果有换行符

var user\_id = "

https://avatars1.githubusercontent.com/u/3949015?v=3&s=40";

// Uncaught SyntaxError: Unexpected token ILLEGAL

**优化：**  
  
可以通过指向的某一个变量存放所有后端返回的内容，最小程度占用全局变量。例：

// PHP 代码

var SERVER\_DATA = {

username: {$username},

userid: {$userid},

title: {$title}

}

// 渲染结果

var SERVER\_DATA = {

username: "NimoChu",

userid: 1,

title: 'F2E'

}

**使用建议：**  
  
需要最快速度传递数据给 JS 并十分确定此数据稳定时，使用此方式。数据格式复杂的建议使用script填充JSON 或AJAX获取JSON 方法。

**script填充JSON**

[什么是JSON？](http://www.w3school.com.cn/json/json_syntax.asp)

填充 JSON 数据到 <script> 标签中，前端通过 DOM 获取 JSON字符串并解析成对象。

<!-- PHP 代码 -->

<script type="text/json" id="data"><?php echo json\_encode($data) ?></script>

<!-- 页面渲染结果 -->

<script type="text/json" id="data">{"username":"nimojs","userid":1}</script>

<script>

var data = JSON.parse($('#data').html());

//{username:"nimojs",userid:1}

</script>

**优点：**  
  
页面加载完成后就可以获取到数据。不占用全局变量，可传递大量数据集合。

**缺点：**  
  
数据量特别大时会导致页面初次加载变慢。变慢并不只是文件大小导致的，也因为服务器查询数据并返回合集是需要时间，可使用AJAX获取JSON完成按需加载和加载等待。

**使用建议：**  
  
适合传递在DOM加载完成时就需要用到的大量数据集合。例如：前端控制页面渲染，后端将JSON数据源填充到 <script> 由前端使用 [JavaScript模板引擎](http://www.gbtags.com/technology/javascript/20120917-javascript-template-engine-chooser/)进行页面渲染。

**AJAX获取JSON**

使用 AJAX 获取JSON数据

<span id="showdata">查看资料</span>

<div style="display:none;" id="box">

<h2>用户信息</h2>

<p id="info"><img src="loading.gif" /></p>

</div>

$('#showdata').on('click',function(){

$('#box').show();

$.getJSON('/ajax/userdata/',function(oData){

// oData = {"username":"nimojs","userid":1}

$('#info').html('用户名：' + oData.username + '<br>用户ID：' + oData.userid);

})

})

这是一个通过AJAX 获取用户资料的示例。流程如下：

1. 页面上只显示查看资料
2. 用户点击查看资料
3. 显示用户信息和 loading 图片
4. 向服务器发送获取用户信息的AJAX请求
5. 服务器返回JSON字符串，$.getJSON 自动将返回的 JSON字符串转换为对象
6. 填充内容到 <p id="info">

**优点：**  
  
不占用全局变量和 DOM 节点，可以自由控制获取数据的触发条件（页面加载完成时、用户点击查看资料时或用户点击某个按钮时）。当开始获取数据时可使用 loading 图片占位提示用户数据正在读取。防止页面加载所有数据导致的页面加载缓慢。

**缺点：**  
  
会产生额外的HTTP请求。不能在DOM加载完成以后立即获取，需要发送请求-接收响应。

**使用建议：**  
  
适合加载非主要信息、设定触发条件（用户点击查看资料时），并提供友好的数据读取等待提示。

**WebSocket实时传输数据**

如果将 AJAX请求和响应比喻成给服务器发短信和等待服务器回复短信，而 WebSocket 就如同和服务器打电话。

此处不对WebSocket做过多介绍，附上参考资料：  
1. [Wiki:WebSocket](http://zh.wikipedia.org/wiki/WebSocket)  
2. [使用 HTML5 WebSocket 构建实时 Web 应用](http://www.ibm.com/developerworks/cn/web/1112_huangxa_websocket/)  
3. [Ajax vs WebSocket](http://www.web-tinker.com/article/20372.html?utm_source=tuicool)

**总结**

每种情况都有每种情况的用处，没有绝对正确的方法。**根据实际情况灵活的选择获取数据方式**。

相关链接

* [知乎：前端 ，后端 关于数据交互的问题?](http://www.zhihu.com/question/26532621)

[点此订阅博客](https://github.com/nimojs/blog/issues/15)

若作者显示不是Nimo（被转载了），请访问Github原文进行讨论：[#13](https://github.com/nimojs/blog/issues/13)