

一篇搞清AJAX及实现手动封装



jCodeLife



0.437 2020.09.07 12:04:25 字数 1,544 阅读 73

编辑文章

所谓 **AJAX** 就是异步 **JavaScript** 和 **XML**，是一种无须加载整个网页就能实现更新部分网页的技术

那如何发送一个 **AJAX** 请求

1. 创建一个异步对象

```
1 var xmlhttp;
2 if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
3     xmlhttp=new XMLHttpRequest();
4 }else { // code for IE6, IE5
5     xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
6 }
```

通过 **XMLHttpRequest** 构造函数创建一个异步对象 **xmlhttp**，IE6, IE5 使用 **ActiveXObject** 创建，创建的这个异步对象上有很多属性和方法，常用的有：

(1) onreadystatechange

监听异步对象请求状态码 **readyState** 的改变，每当 **readyState** 改变时，就会触发 **onreadystatechange** 事件

(2) readyState：请求状态码

readyState 表示异步对象目前的状态，状态码从0到4：

0：表示请求未初始化，还没有调用 **open()**

1：服务器连接已建立，但是还没有调用 **send()**

2：请求已接收，正在处理中（通常现在可以从响应中获取内容头）

3：请求处理中，通常响应中已有部分数据可用了，没有全部完成

4：当**readyState**状态码为4时，表示请求已完成；此阶段确认全部数据都已经解析完毕，可以通过异步对象的属性获取对应数据

(3) status：http状态码

http状态码表示成功的http状态码有 **xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp.status == 304**

(4).responseText：后台返回的字符串形式的响应数据

(5) responseXML：后台返回的XML形式的响应数据

2. 设置请求方式和请求地址

创建异步对象之后，通过 **open()** 方法设置 **ajax** 请求方式和请求地址

格式：**xmlhttp.open("GET/POST", "ajax-get.txt", true);**

第一个参数：请求的类型：**GET** 还是 **POST**

第二个参数：表示请求的文件的地址 **url**

第三个参数：设置请求方法是不是异步 **async**，**true** 为异步，**false** 为同步。**AJAX** 存在的意义就是发异步请求，所以第三个参数永远传 **true**

这里有个问题，就是 **IE** 中的缓存问题

在 **IE** 浏览器中，如果通过 **Ajax** 发送 **GET** 请求，那么 **IE** 浏览器认为，同一个URL只有一个结果，如果地址没有发生变化，它就会把上一次返回的结果，直接返回。这样我们不能实时的拿到变化后的数据。如果要想我们拿到实时数据，必须保证每次的URL都是不一样的，有两种方式：

(1) Math.random()：随机数

推荐阅读

如何理解JavaScript中的对象？

阅读 246

js 判断字符串中是否包含某个字符串

阅读 235

从事7年前端开发，有些经验想对转行学习前端的伙伴说说！

阅读 175

Vue插件-饼状图

阅读 819

computed、watch和methods区别

阅读 100

```
1 xmlhttp.open("GET","ajax-get.txt?t=" + (new Date().getTime()),true);
2 //或
3 xmlhttp.open("GET","ajax-get.txt?t=" + Math.random(),true);
```

3. 发送请求

直接通过异步对象的send()发送请求

```
1 | xmlhttp.send();
```

特别注意的是：如果发送 **POST** 请求，请使用 `setRequestHeader()` 来添加 **HTTP** 请求头，并在 `send()` 方法中传递要发送的数据：

```
1 xmlhttp.open("POST","ajax_test.html",true);
2 xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
3 xmlhttp.send("fname=Henry&lname=Ford");
```

4. 通过onreadystatechange监听状态变化

每当异步对象的 `readyState` 发生改变，就会触发 `onreadystatechange` 函数，当 `readyState` 变成 **4** 时，表示当前状态是请求完毕的状态，同时当 `http` 的响应码 `status` 为 **200** 到 **300** 之间（包括 **200** 和 **300**）或为 **304** 时，表示 `ajax` 请求成功；当 `http` 状态码不是 **200** 到 **300** 之间的数也不是 **304** 时，表示请求不成功

```
1 //4. 监听状态变化
2 xmlhttp.onreadystatechange = function(){
3   // 判断当前状态改变是请求完毕的状态吗
4   if (xmlhttp.readyState === 4) {
5     if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp.status == 304) {
6       console.log("成功的接收到服务器返回的数据");
7     }else{
8       console.log("不成功！");
9     }
10  }
11 }
```

5. 处理返回的结果

如果成功，可通过异步对象的 `responseText` 属性来获取服务器返回的字符串

```
1 | console.log(xmlhttp.responseText);
```

根据上述情况，如果每次请求都要手写以上五个步骤，那很麻烦，接下来，我们来封装一个方法 `ajax()` 用于发送请求

封装的时候，需要注意

(1) **URL** 当中只能出现字母 数字 下划线和 **ASCII** 码，不能出现中文，可以使用

`encodeURIComponent()` 转码

(2) 当我们利用我们的 `ajax` 放的发送一个请求到远处服务器时，我们需要等待远程服务器去响应我们的请求，等待远程服务器将响应的结果返回给我们，但是这个响应的速度是不确定的，因为响应的速度是由本地网络和远程服务器的网速等共同决定的，所以我们不可能一直等待服务器的响应。那么这里我们需要新增另外一个功能：设置超时时间的功能，即告诉它我的请求会等待多长的时间，如果这么长的时间内都没有响应我们发送的请求，那我就在这里自动的终

推荐阅读

如何理解JavaScript中的对象？

阅读 246

js 判断字符串中是否包含某个字符串

阅读 235

从事7年前端开发，有些经验想对转行学习前端的伙伴说说！

阅读 175

Vue插件-饼状图

阅读 819

computed、watch和methods区别

阅读 100

```
3  var str = objToString(obj);
4  // 1.创建一个异步对象xmlhttp:
5  var xmlhttp, timer;
6  if (window.XMLHttpRequest) {
7      xmlhttp = new XMLHttpRequest();
8  } else { // code for IE6, IE5
9      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
10 }
11 // 2.设置请求方式和请求地址;
12 // 判断请求的类型是POST还是GET
13 if (type === 'GET') {
14     xmlhttp.open(type, url + "?t=" + str, true);
15     // 3.发送请求;
16     xmlhttp.send();
17 } else {
18     //如果是post请求, 需要设置请求头
19     xmlhttp.open(type, url, true);
20     // 注意: 在post请求中, 必须在open和send之间添加HTTP请求头: setRequestHeader(header,value);
21     xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
22     // 3.发送请求;
23     xmlhttp.send(str);
24 }
25 // 4.监听状态的变化:
26 xmlhttp.onreadystatechange = function () {
27     clearInterval(timer);
28     if (xmlhttp.readyState === 4) {
29         if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp.status == 304) {
30             // 5.处理返回的结果;
31             success(xmlhttp); //成功后回调;
32         } else {
33             error(xmlhttp); //失败后回调;
34         }
35     }
36 }
37 //处理obj
38 function objToString(obj) {
39     obj.t = new Date().getTime();
40     var res = [];
41     for (var key in obj) {
42         //需要将key和value转成非中文的形式, 因为url不能有中文。使用encodeURIComponent();
43         res.push(encodeURIComponent(key) + "=" + encodeURIComponent(obj[key]));
44     }
45     return res.join("&");
46 }
47 //判断外界是否传入了超时时间
48 if (timeout) {
49     timer = setInterval(function () {
50         xmlhttp.abort(); //中断请求
51         clearInterval(timer);
52     }, timeout);
53 }
54 }
```

在使用自己封装的代码的时候, 感觉跟 `jQuery` 官方的 `ajax` 还是有一定的差异, 所以还需要进一步完善

首先看第一个问题, 传递多个参数, 需要保持传递顺序。解决方案是可以改写成传递的是一个对象; 因为对象里面的值, 传递的是一个对象就不用考虑先后顺序, 里面用的参数通过 `对象名.属性名` 的形式获取

第二, 传递请求类型的区分大小写, `jQuery` 官方的是大小写都可以; 解决方案是可以使用 `toLowerCase()` 或者 `toUpperCase()` 将类型转成大写或小写再对比;

第三, 我们传递的数据用的名字是 `obj`, `jQuery` 官方用的是 `data`, 语义更适合。所以将传递数据 `obj` 改成 `data`

完善如下:

```
1 function ajax(option) { //type,url,obj,timeout,success,error将所有参数换成一个对象{}
```

推荐阅读

如何理解JavaScript中的对象?

阅读 246

js 判断字符串中是否包含某个字符串

阅读 235

从事7年前端开发, 有些经验想对转行学习前端的伙伴说说!

阅读 175

Vue插件-饼状图

阅读 819

computed、watch和methods区别

阅读 100

```
8      } else { // code for IE6, IE5
9          xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
10     }
11     // 2. 设置请求方式和请求地址;
12     // 判断请求的类型是POST还是GET
13     if (option.type.toLowerCase() === 'get') {
14         xmlhttp.open(option.type, option.url + "?t=" + str, true);
15         // 3. 发送请求;
16         xmlhttp.send();
17     } else {
18         xmlhttp.open(option.type, option.url, true);
19         // 注意: 在post请求中, 必须在open和send之间添加HTTP请求头: setRequestHeader(header,value);
20         xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
21         // 3. 发送请求;
22         xmlhttp.send(str);
23     }
24     // 4. 监听状态的变化;
25     xmlhttp.onreadystatechange = function () {
26         clearInterval(timer);
27         if (xmlhttp.readyState === 4) {
28             if (xmlhttp.status >= 200 && xmlhttp.status < 300 || xmlhttp.status == 304) {
29                 // 5. 处理返回的结果;
30                 option.success(xmlhttp); // 成功后回调;
31             } else {
32                 option.error(xmlhttp); // 失败后回调;
33             }
34         }
35     }
36     // 处理obj
37     function objToString(data) {
38         data.t = new Date().getTime();
39         var res = [];
40         for (var key in data) {
41             // 需要将key和value转成非中文的形式, 因为url不能有中文. 使用encodeURIComponent();
42             res.push(encodeURIComponent(key) + "=" + encodeURIComponent(data[key]));
43         }
44         return res.join("&");
45     }
46     // 判断外界是否传入了超时时间
47     if (option.timeout) {
48         timer = setInterval(function () {
49             xmlhttp.abort(); // 中断请求
50             clearInterval(timer);
51         }, timeout);
52     }
53 }
```

最后传参形式变成

```
1 ajax({
2     type: xxx,
3     url: xxx,
4     data: xxx,
5     timeout: xxx,
6     success: xxx,
7     error: xxx
8 });
```

以上就是ajax相关的内容, 基本功能是能实现, 但跟jQuery的ajax还是差很多, 考虑的也不全, 想了解jQuery的ajax可以阅读以其源码, 后面有时间我也会一起来看一下



4人点赞 >



笔记 (2)



"小礼物走一走, 来简书关注我"

赞赏支持

推荐阅读

如何理解JavaScript中的对象?

阅读 246

js 判断字符串中是否包含某个字符串

阅读 235

从事7年前端开发, 有些经验想对转行学习前端的伙伴说说!

阅读 175

Vue插件-饼状图

阅读 819

computed、watch和methods区别

阅读 100

写下你的评论...

评论 0

赞 4

...



写下你的评论...



Ctrl + Enter 发表

发布

取消

全部评论 0

只看作者

关闭评论

按时间倒序

按时间正序

被以下专题收入，发现更多相似内容

投稿管理

+ 收入我的专题



前段开发那些事儿

推荐阅读

如何理解JavaScript中的对象?

阅读 246

js 判断字符串中是否包含某个字符串

阅读 235

从事7年前端开发，有些经验想对转行学习前端的伙伴说说!

阅读 175

Vue插件-饼状图

阅读 819

computed、watch和methods区别

阅读 100



写下你的评论...



评论0



赞4

