

AJAX工作原理

github.com/tsrot

JavaScript闭包

地址：<http://blog.xieliquan.com/2016/08/08/ajax/>

AJAX 是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

一、ajax所包含的技术

大家都知道ajax并非一种新的技术，而是几种原有技术的结合体。它由下列技术组合而成。

- 使用CSS和XHTML来表示。
- 使用DOM模型来交互和动态显示。
- 使用XMLHttpRequest来和服务器进行异步通信。
- 使用javascript来绑定和调用。

在上面几种技术中，除了XmlHttpRequest对象以外，其它所有的技术都是基于web标准并且已经得到了广泛使用的，XMLHttpRequest虽然目前还没有被W3C所采纳，但是它已经是一个事实的标准，因为目前几乎所有的主流浏览器都支持它。

二、怎样创建ajax

Ajax的原理简单来说通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用javascript来操作DOM而更新页面。这其中最关键的一步就是从服务器获得请求数据。原生创建ajax可分为以下四步。

1、创建XMLHttpRequest对象

所有现代浏览器（IE7+、Firefox、Chrome、Safari 以及 Opera）均内建XMLHttpRequest对象。创建XMLHttpRequest对象的语法：

```
var xhr = new XMLHttpRequest();
```

老版本的Internet Explorer（IE5和IE6）使用ActiveX对象：

```
var xhr = new ActiveXObject("Microsoft.XMLHTTP");
```

为了应对所有的现代浏览器，包括 IE5 和 IE6，请检查浏览器是否支持 XMLHttpRequest 对象。如果支持，则创建 XMLHttpRequest 对象。如果不支持，则创建 ActiveXObject：

```
var xhr;
if(XMLHttpRequest){
    xhr = new XMLHttpRequest();
}else{
    xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
```

2、准备请求

初始化该XMLHttpRequest对象，接受三个参数：

```
xhr.open(method,url,async);
```

第一个参数表示请求类型的字符串，其值可以是GET或者POST。

GET请求：

```
xhr.open("GET",demo.php?name=tsrot&age=24,true);
```

POST请求：

```
xhr.open("POST",demo.php,true);
```

第二个参数是要作为请求发送目标的URL。

第三个参数是true或false，表示请求是以异步还是同步的模式发出。（默认为true，一般不建议为false）

- false：同步模式发出的请求会暂停所有javascript代码的执行，知道服务器获得响应为止，如果浏览器在连接网络时或者在下载文件时出了故障，页面就会一直挂起。
- true：异步模式发出的请求，请求对象收发数据的同时，浏览器可以继续加载页面，执行其他javascript代码

3、发送请求

```
xhr.send();
```

一般情况下，使用Ajax提交的参数多是些简单的字符串，可以直接使用GET方法将要提交的参数写到open方法的url参数中，此时send方法的参数为null或为空。

GET请求：

```
xhr.open("GET",demo.php?name=tsrot&age=24,true);
xhr.send(null);
```

POST请求：

如果需要像 HTML 表单那样 POST 数据，请使用 `setRequestHeader()` 来添加 HTTP 头。然后在 `send()` 方法中规定您希望发送的数据：

```
xhr.open("POST",demo.php,true);
xhr.setRequestHeader("Content-Type","application/x-www-form-urlencoded; charset=UTF-8");
xhr.send("name="+userName+"&age="+userAge);
```

4、处理响应

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

`onreadystatechange` 事件：

当请求被发送到服务器时，我们需要执行一些基于响应的任务。每当 `readyState` 改变时，就会触发 `onreadystatechange` 事件。

`readyState`属性：

- 0：已经创建对象，但还没有调用`open()`方法。
- 1：已经调用`open()`方法，但还没有发送请求。
- 2：请求已经发送，标题和状态已经收到，并可用。
- 3：接收到来自服务器的响应。
- 4：接收完请求数据，表示已经完成请求。

`status`属性：

200:"OK"

404: 未找到页面

`responseText`：获得字符串形式的响应数据

`responseXML`：获得 XML 形式的响应数据

返回值一般为json字符串，可以用`JSON.parse(xhr.responseText)`转化为JSON对象。

5、完整例子

demo.html

```

var xhr;
if(XMLHttpRequest){
    xhr = new XMLHttpRequest();
}else{
    xhr = new ActiveXObject("Microsoft.XMLHTTP");
};

xhr.open("GET","./data.json",true);
xhr.send();

xhr.onreadystatechange = function(){
    if(xhr.readyState == 4 && xhr.status == 200){
        console.log(JSON.parse(xhr.responseText).name);
    }
}

```

data.json

```

{
    "name":"tsrot",
    "age":24
}

```

三、ajax应用场景

- 场景 1. 数据验证
- 场景 2. 按需取数据
- 场景 3. 自动更新页面

四、ajax优缺点

优点：

- 1、页面无刷新，用户体验好。
- 2、异步通信，更加快的响应能力。
- 3、减少冗余请求，减轻了服务器负担
- 4、基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

缺点：

- 1、ajax干掉了back按钮，即对浏览器后退机制的破坏。
- 2、存在一定的安全问题。
- 3、对搜索引擎的支持比较弱。
- 4、破坏了程序的异常机制。
- 5、无法用URL直接访问。