

10 分钟快速解读 Ajax 技术

1. 同步和异步的区别是什么？

同步：

浏览器向服务器请求数据，服务器比较忙，浏览器一直等着（页面白屏），直到服务器返回数据，浏览器才能显示页面。

异步：

浏览器向服务器请求数据，服务器比较忙，浏览器可以自如的干原来的事情（显示页面），服务器返回数据的时候通知浏览器一声，浏览器把返回的数据再渲染到页面，局部更新。

2. 隐藏帧 iframe 方式实现页面局部更新

虽然只是 iframe 子页面进行了刷新，从而实现了页面的局部更新，但是没有通过异步来实现，也就是说通过 iframe 进行页面局部更新的时候，是页面同步更新，当服务器比较忙的时候，浏览器一直等着（页面白屏），直到服务器返回数据，浏览器才能显示页面。

3. Ajax 主要用来实现客户端与服务器端的异步通信效果，实现页面的局部刷新

在不刷新页面的情况下，浏览器悄悄地、异步地向服务器发出 HTTP 请求。服务器收到请求后，传回新的格式化数据回来（通常是 JSON）。浏览器解析 JSON，通过 DOM 将新数据呈递显示，页面仅局部刷新。

4. 简述 ajax 的过程

```
// 1.创建 XMLHttpRequest 对象,也就是创建一个异步调用对象
var xhr = null;//创建对象
if(window.XMLHttpRequest){
    xhr = new XMLHttpRequest();
}else{
    xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
// 2. 设置发送的参数（请求方式，请求的地址，同步或者异步标识位）
xhr.open('get/post', 'url', 'async');
// 3.发送 HTTP 请求
xhr.send();//发送请求
// 4.指定回调函数，获取异步调用返回的数据，使用 JS 和 DOM 实现局部刷新
xhr.onreadystatechange =function(){}
```

5. 同源与跨域的区别？

所谓同源指的是请求 URL 地址中的协议、域名和端口号都相同，只要其中之一不相同就是跨域。

跨域是指从一个域名的网页去请求另一个域名的资源。只要协议、域名、端口有任何一个的不同，就被当作是跨域。

6. 同源策略是什么？

同源策略是浏览器的一种安全策略，同源策略主要是为了保证浏览器的安全性，所以在同源策略下，浏览器不允许 Ajax 跨域获取服务器数据。

7. 跨域解决方案之一 —— jsonp（99%的公司都在使用 jsonp）

由于<script>标签中的 src 没有跨域限制，所以 jsonp 充分利用了这一特性，动态添加<script>标签来调用服务器提供的 js 脚本。

8. jsonp 的原理

动态创建 script 标签，通过它的 src 属性发送跨域请求，然后服务器响应的数据格式为函数调用，并且通过函数调用，来传递一些参数，所以在发送请求之前必须先声明一个函数，并且函数的名字与参数中传递的名字要一致，这样服务器响应回来的回调函数就可以直接调用这里声明的函数，从而实现实参（数据）的传递，再通过 JS 和 DOM 实现局部刷新。

备注：为什么要返回一个回调函数？

因为动态创建 script 标签发出去请求是异步请求，直接从服务器端返回数据的话，我们获取不到，自然就不能直接使用返回的数据。

9. Ajax 与 jsonp 本质区别

Ajax 的核心是通过 XMLHttpRequest 获取非本页内容，本质是同源的（url 地址具有唯一性，当然，地址后面可以通过 “?键=值&键=值” 或者 “:ip” 等方式传递参数）；

jsonp 的核心则是动态添加<script>标签来调用服务器提供的 js 脚本，本质是可以跨域请求数据。

10. jQuery 中封装的 ajax 方法

由于实际开发网站的过程中，这两种技术很可能交替使用，因此 jQuery 把 Ajax 与 jsonp 两种方法进行了合并式的封装，当我们调用 jQuery 中的 ajax 方法的时候，我们必须熟练掌握的参数如下：

```

$(function() {

    $.ajax({

        type: 'get', // 请求方式: 'get' / 'post' 必须得写

        async: true, // 默认是 async: true, 异步请求数据, 所以可以省略不写

        url: '#', // url 地址, 这是后台接口 (是由后台提供给我们的),
                // 只有通过 url 才可以访问到服务器中对应的我们要请求的资源

        data: {}, // 根据接口文档的要求, 如果需要我们传入对应的参数 (例如: name:String)
                // 我们就以键值对的方式, 将参数依次传入

        dataType: 'text', // dataType: 'text / json', 说明是 ajax 方法调用
                // dataType: 'jsonp', 说明是 jsonp 方法调用

        jsonp: 'callback', // 当 dataType: 'jsonp', 必须要和后台沟通并填写 jsonp 属性
                // jsonp 属性的作用就是自定义参数名字, jquery 的默认参数名称是 callback
                // 后端根据这个参数名字获取回调函数的名字 (方法名)

        jsonpCallback: '123r87ebdfbdgdf839fer1455623', // 这个属性的作用就是自定义回调函数的名字,
                // jQuery 中默认是一个随机数

        success: function( data ) {

            console.log( data );
            //根据返回的 data 数据, 进行 js 、Dom 等操作, 将数据渲染到页面, 实现局部刷新

        }

    });

});

```