

## 1、首先引出问题，到底什么是跨域？

跨域，指的是浏览器不能执行其他网站的脚本。

举例：

比如在网站中使用 ajax 请求其他网站的天气、快递或者其他数据接口

## 2、产生跨域的原因

它是由浏览器的同源策略造成的，是浏览器对 javascript 施加的安全限制。

引出什么是同源策略？

同域名，同端口，同协议，举个例子：

http://www.123.com/index.html 调用 http://www.123.com/server.php （非跨域）

http://www.123.com/index.html 调用 http://www.456.com/server.php （主域名不同:123/456，跨域）

http://abc.123.com/index.html 调用 http://def.123.com/server.php （子域名不同:abc/def，跨域）

http://www.123.com:8080/index.html 调用 http://www.123.com:8081/server.php （端口不同:8080/8081，跨域）

http://www.123.com/index.html 调用 https://www.123.com/server.php （协议不同:http/https，跨域）

## 3、使用什么方法可以解决跨域？

### a) CORS 跨域资源共享

HTML5 中提供的 XMLHttpRequest Level2（及 XHR2）已经实现了跨域访问。但 ie10

以下不支持。只需要在服务端填上响应头：

header("Access-Control-Allow-Origin:");/\*星号表示所有的域都可以接受，\*/

**优点：**代表任何域。也可以指定地址

Access-Control-Allow-Methods: POST,GET //支持的方法

**缺点：**ie10 以下不支持

**解决方法：**可以用 XDR 实现，XDR 是 IE8、IE9 提供的一种跨域解决方案，但是功能较弱只支持 get 跟 post 请求，而且对于协议不同的跨域是无能为力的（比如在 http 协议下发送 https 请求）

### b) 动态创建 script 标签——只支持 GET 请求

由于 script/img/link 标签的 src 属性都不受同源策略的影响，所以可以动态创建 script 标签，代码如下：

```
var script = document.createElement('script');
script.src = "http://localhost:8080/kuayu2/test.php";
document.body.appendChild(script);
```

使用 jQuery 的 jsonp 请求

```
$("#btn").click(function() {
    $.ajax({
        url:"http://localhost:8080/kuayu3/test.php",
        type:"get",
        dataType:'jsonp', //定义要发送的 jsonp 请求
        jsonp:'kyFn', //更改定义的参数名
    });
});
```

```

        jsonpCallback:'hyly', //指定 jsonp 发送的回调函数名
        success:function(info) {
            console.log(info);
        }
    });
})

```

服务器端脚本:

```

<?php
    header("Content-type: text/html;charset=utf-8");
    $callback=$_GET['kyFn'];
    echo $callback.'("哈哈")';
?>

```

#### 优点:

它不像 XMLHttpRequest 对象实现的 Ajax 请求那样受到同源策略的限制;它的兼容性更好,在更加古老的浏览器中都可以运行,不需要 XMLHttpRequest 或 ActiveX 的支持;并且在请求完后可以通过调用 callback 的方式回传结果

#### 缺点:

只能用 get 方法,不能使用 post 方法

#### c) 基于 iframe 实现跨域:

##### 应用场景:

基于 iframe 实现的跨域要求两个域具有 aa.xx.com, bb.xx.com 这种特点,也就是两个页面必须属于一个基础域(例如都是 xxx.com,或是 xxx.com.cn),使用同一协议(例如都是 http)和同一端口(例如都是 80),这样在两个页面中同时添加 document.domain,就可以实现父页面调用子页面的函数

**缺点:** 只有在主域名相同的情况下才能使用该方法

#### d) web sockets 解决跨域:

**原理:** 在 JS 创建了 web sockets 之后,会有一个 HTTP 请求发送到浏览器以发起链接,取得服务器响应后,建立的连接会使用 HTTP 升级从 HTTP 协议交换为 web sockets 协议

**缺点:** 只有在支持 web sockets 协议的服务器上才能正常工作