

Ajax 入门和应用

课堂笔记 2

目录

一、JSONP 跨域.....	2
1.1 JSONP 概念.....	2
1.2 JSONP 实用函数封装.....	5

一、JSONP 跨域

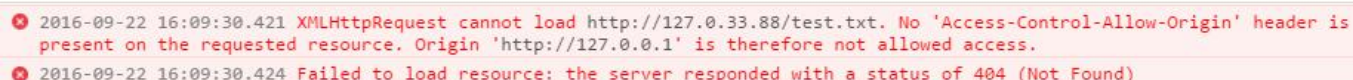
1.1 JSONP 概念

Ajax 不能跨域,比如你的页面在 127.0.0.1 里,不能请求 127.0.0.2 里面的 JSON 文件。比如,你在 www.163.com 里,不能请求 www.sohu.com 里面的 JSON 文件。

域名不同 (实际上 ip 地址,端口号不同也不行), Ajax 不能跨域名的。比如 news.sina.com.cn 能不能请求 ent.sina.com.cn 不一定! 如果 news 这个机器和 ent 是一个 ip, 就能, 如果不是一个机器, 就不能。

比如我们现在的页面运行在 127.0.0.1, 试图请求 127.0.33.88 的文件, 会报错:

```
$.get("http://127.0.33.88/test.txt",function(data){  
    alert(data);  
});
```



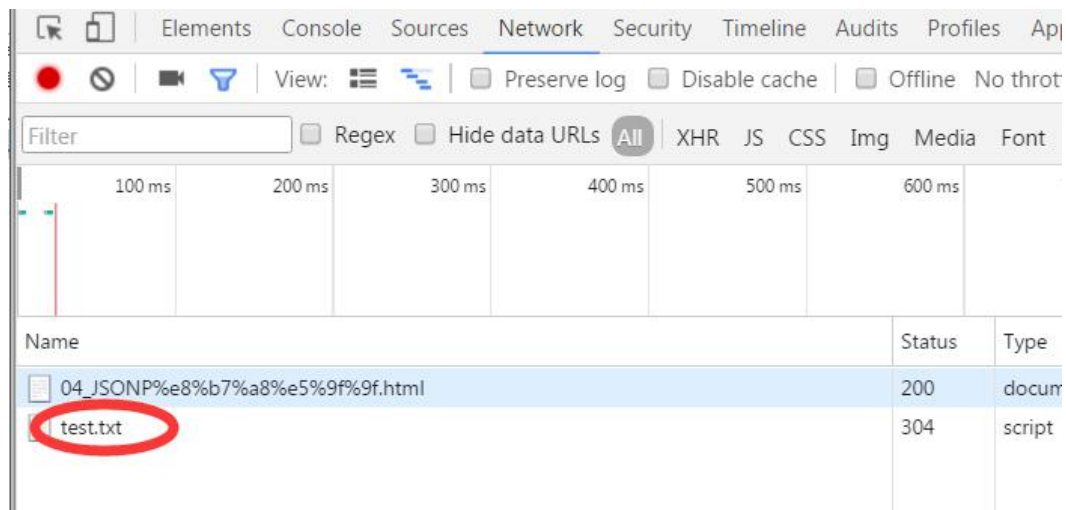
你觉得不能跨域这个事情, 是浏览器的限制, 还是服务器的限制? 是服务器的限制, 服务器有机制阻止来自其他网址的 XHR 请求。

比如 A 服务器上运行着页面, B 服务器上有数据库 (有 JSON), 现在 A 必须去请求 B, 怎么办? 用 JSONP。

JSONP 不是 Ajax! 是一种技巧, 发明人不可考, 近些年变得流行。

首先, 我们写一个 script 标签, 这个标签现在有一个 src 属性, 指向一个 txt 文件。此时将会发出 HTTP 请求, 请求这个文件。

```
<script type="text/javascript" src="test.txt"></script>
```



我们现在请求不变：

```
<script type="text/javascript" src="test.txt"></script>
```

但是改变文件

```
alert("你好");
```

此时你好就能弹出，这是因为 script 标签会引包，同时也执行了 js 文件。

问题是，**HTML 页面没有变量能够持有这个你好两个字**，所以我们现在就必须用技巧：

```
fun("你好");
```

我们之前都是在 js 文件中定义一个函数，在主页面中调用；而今天我们给反过来了！！在主页面定义函数，在 js 文件中调用函数！！

主页面中定义函数：

```
function fun(obj){  
    alert(obj.result[1].nianling);  
}
```

在外部文件中调用函数：

```
fun("你好");
```

此时外部文件的数据，被当做了实参传入了 HTML 页面，被形参接收。

如果参数是 JSON 呢？那 JSON 不就进页面了么！！我们起了一个名字，JSON with Padding。简称 JSONP。

红色的部分（调用函数语句）就是所谓的 padding。

```
fun(  
    {  
        "result" : [  
            {  
                "xingming":"小明",  
                "nianling" : 12,  
                "sex" : "男"  
            },  
            {  
                "xingming":"小红",
```

```

        "nianling" : 13,
        "sex" : "女"
    },
    {
        "xingming":"小绿",
        "nianling" : 9,
        "sex" : "女"
    }
]
}

```

);

JSONP 是可以跨域的，垫付了 Ajax 的写法。甚至不用出现 XMLHttpRequest 对象就能实现 Ajax。

JSONP 刚刚诞生的时候，整个世界觉得被戏弄了。

JSONP 是函数的调用，通过实参把数据传递到 HTML 页面中。

```

<script type="text/javascript">
    function fun(obj){ → 定义函数

    }
</script>

<script type="text/javascript" src="test.txt"></script> → 调用函数

```

JSONP 不安全。因为如果你的 JSONP 文件，被人篡改了，你的页面就等于被黑了~~。

所以使用 JSONP 的时候，如果 JSONP 文件是自己公司的，放心大胆用，一点毛病没有！

1.2 JSONP 实用函数封装

我们现在能不能封装一个类似：

```

myajax.get(URL,{},function(){
});

```

这样的函数。

主页面:

```
myjsonp("http://www.jd.com/sadfadsf/32rwe/wer",function(data){  
    //data 就是 JSONP 请求回来的值  
}, "fetchJSON_comment98vv21142");
```

JSONP 外面的文件在调用函数,

```
fetchJSON_comment98vv21142 ("你好");
```

这是我们已经写好的函数:

```
function myjsonp(URL,callback,callbackname){  
    //给系统中创建一个全局变量, 叫做 callbackname, 指向 callback 函数  
  
    //定义  
    window[callbackname] = callback;  
  
    //创建一个 script 节点  
    var oscript = document.createElement("script");  
  
    //和 image 不一样, 设置 src 并不会发出 HTTP 请求  
    oscript.src = URL;  
    oscript.type = "text/javascript";  
  
    //script 标签的请求是在上树的时候发出, 请求的是一个函数的执行语句  
    document.head.appendChild(oscript);  
  
    //为了不污染页面, 我们瞬间把 script 拿掉  
    document.head.removeChild(oscript);  
}
```