## Jsonp（重难点）

重要程度：\*\*\*\*\*

### 什么是jsonp？

Jsonp：json with packing的简写。

将json数据包装起来并返回的一种方式 ----- 协议

需要将jsonp和json区分开：

* Json是字符串，是数据
* Jsonp是协议，是传递json数据的方式

Json和jsonp的关系，好比是情报和传递情报的方式。

在使用jsonp实现跨域的时候，它是需要将json数据作为函数的参数（packing的过程），最终在服务端将包装好的字符串作为结果返回给浏览器端。

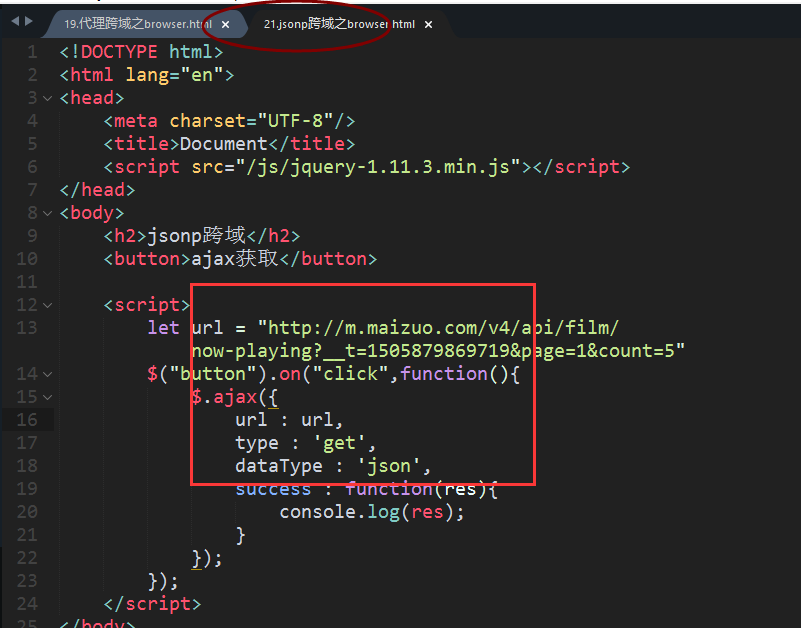
使用场景

在使用jsonp的时候，需要确保：

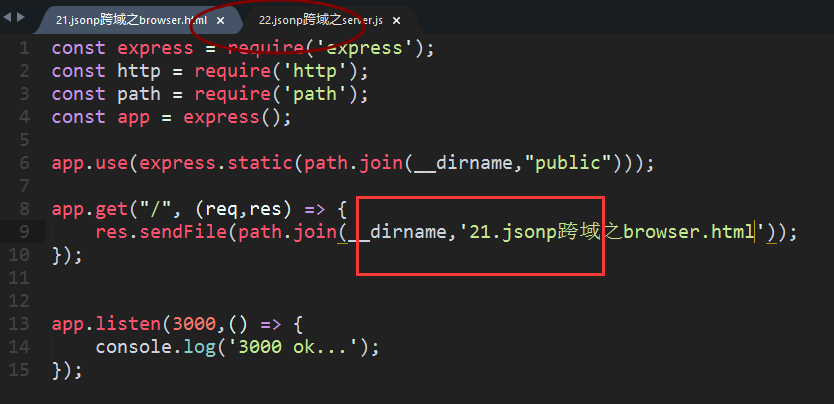
* 有权限去编写服务端的代码
* 在浏览器端也需要写一些代码

### 使用jsonp实现跨域之问题

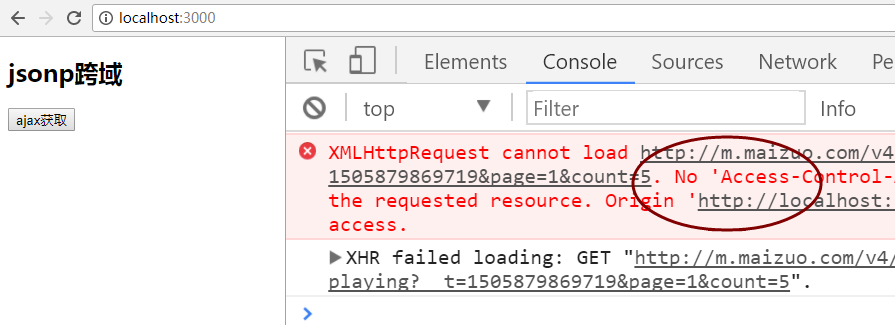
第一步，准备静态页面



第二步，提供一个http server，载入静态页面



浏览，请求，如下：

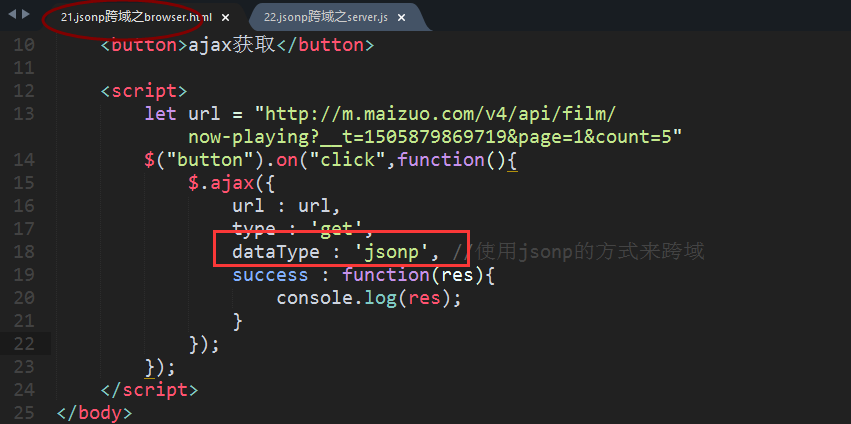


有跨域限制。

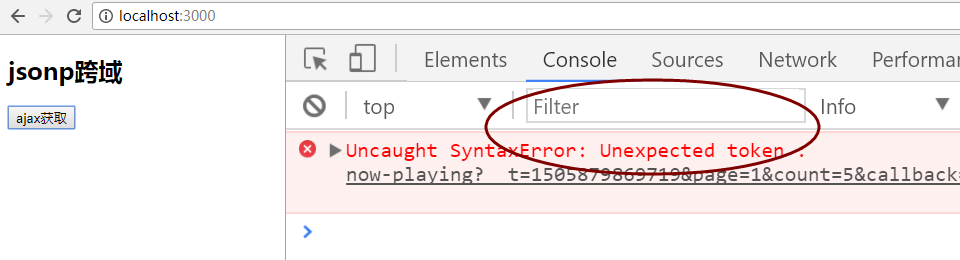
第三步，使用jsonp来实现跨域。

jQuery中的ajax方法，针对jsonp有一个封装。通过dataType来设置的。

只需要将dataType设置为jsonp，如下：

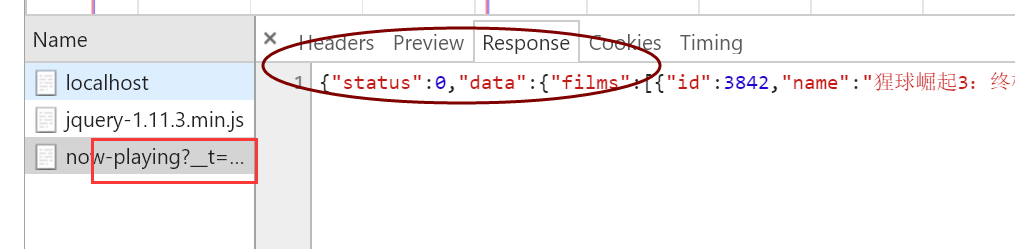


再次访问，如下：

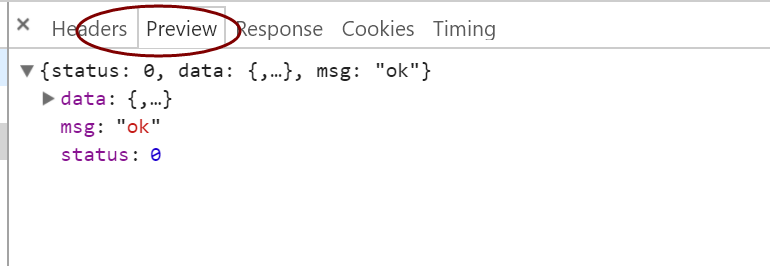


说明有语法错误，这是咋回事呢？

需要通过network分析http协议，我们发现请求已经返回了相应的数据，如下：



分析，返回的结果是一个json字符串，不是包装好的jsonp协议的字符串。在浏览器端，它是按照jsonp的协议来解析这个结果，所以语法错误。



Jsonp返回的不能是json字符串，必须是函数调用的字符串。Json数据作为函数的参数。

如 f1( “{ a : 100, b : 200 }” )

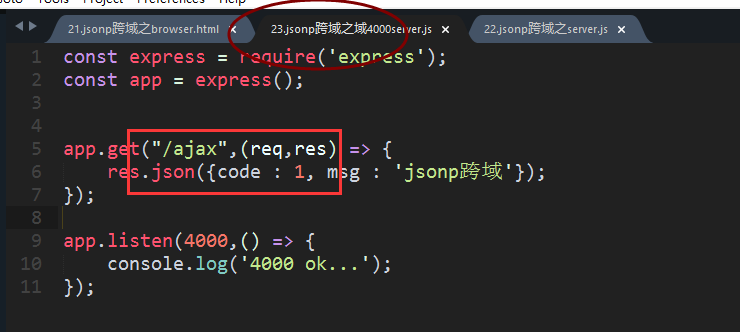
这就需要我们在服务端，返回数据的时候，必须遵循jsonp的协议。

此时，就需要去编写服务端的代码。

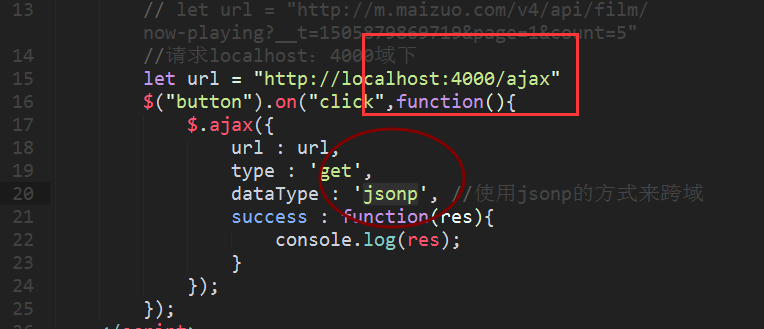
### 使用jsonp实现跨域之解决

要实现jsonp，关键是服务端，必须要以jsonp的协议返回数据。

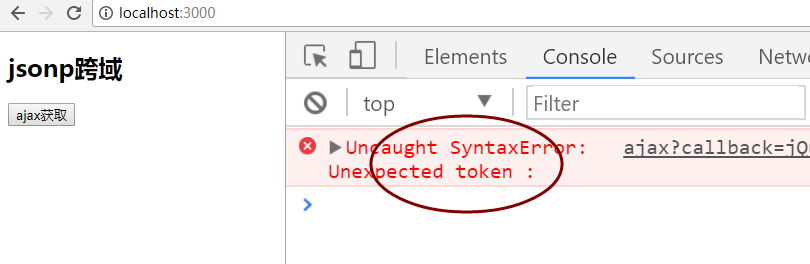
在localhost:4000中提供一个http服务， 如下：



在浏览器中，向localhost:4000/ajax发请求，如下：



启动服务，再次请求，如下：



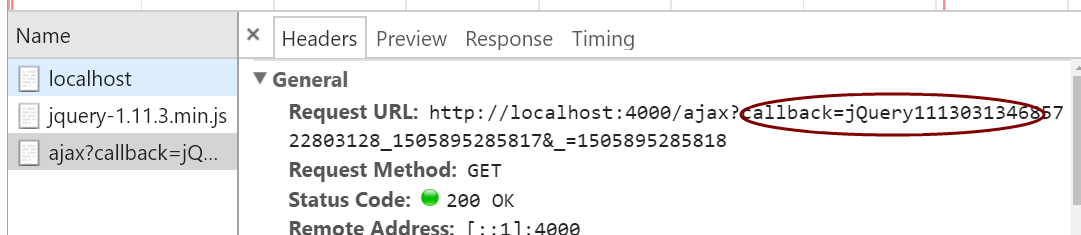
再次，说明，服务端返回的格式是错误，从而导致解析出错。

关键：服务端，必须要返回一个函数调用的字符串，json数据（字符串）作为函数的参数。

请问：

* 函数名是什么？
* 怎么获取的？

再一次使用network分析http，如下：

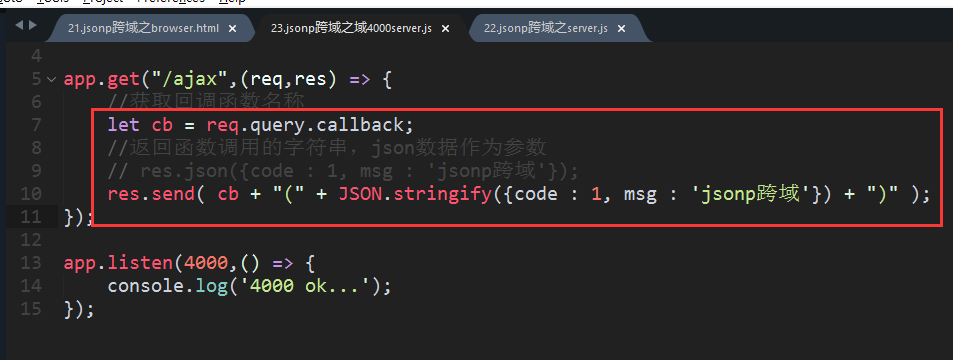


明明请求的路径是/ajax，我也没有传递任何的data，

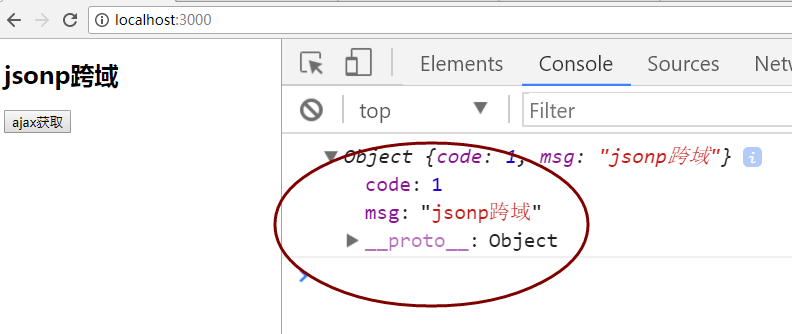


实际上，是由于我们设置了dataType为jsonp，所以jQuery就自己随机了一个函数名，将其作为查询字符串传递给服务端了。

在服务端，通过查询字符串来获取即可。编写代码如下：



重启，测试，结果如下：



小结：

在使用jquery的ajax方法实现jsonp跨域，必须：

* 在浏览器端，设置dataType为jsonp
* 在服务端，必须以函数调用json字符串的形式返回作为字符串返回

在express中，针对jsonp格式的返回，专门封装了一个jsonp方法，如下：



这种使用jsonp方法时候，无需获取函数名，无需转换json，只需要调用jsonp方法，将json数据作为参数即可，如下：



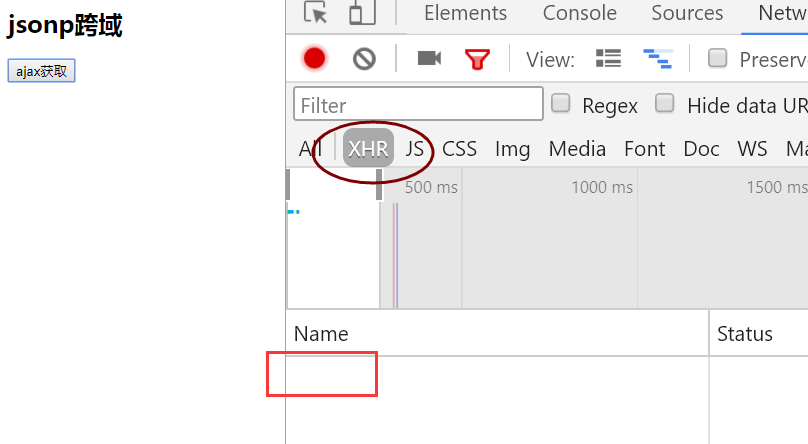
测试，ok。

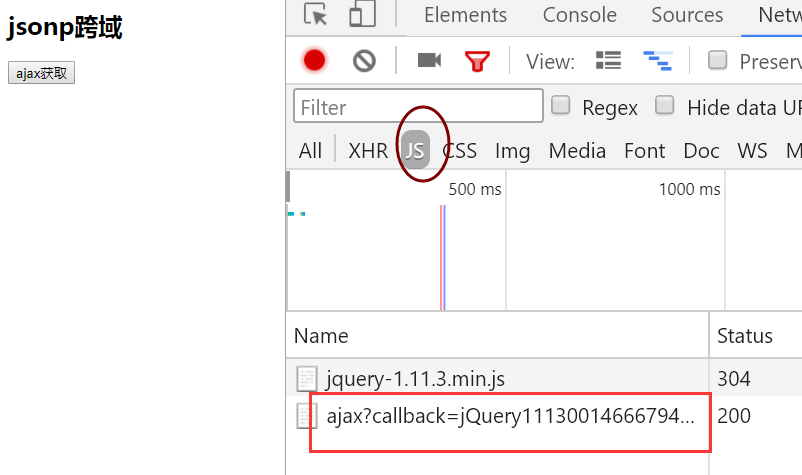
实际上，jsopn方法底层就实现了刚才的那两行代码。

### jsonp原理剖析

由于jQuery封装好了jsonp的用法，导致很多初学者都错误的理解了jsonp。

实际上，jsonp方法并没有使用xhr对象来完成异步请求。查看network，如下：





Jsonp的实现是通过动态的添加script标签从而发送http请求来实现的。

实现如下

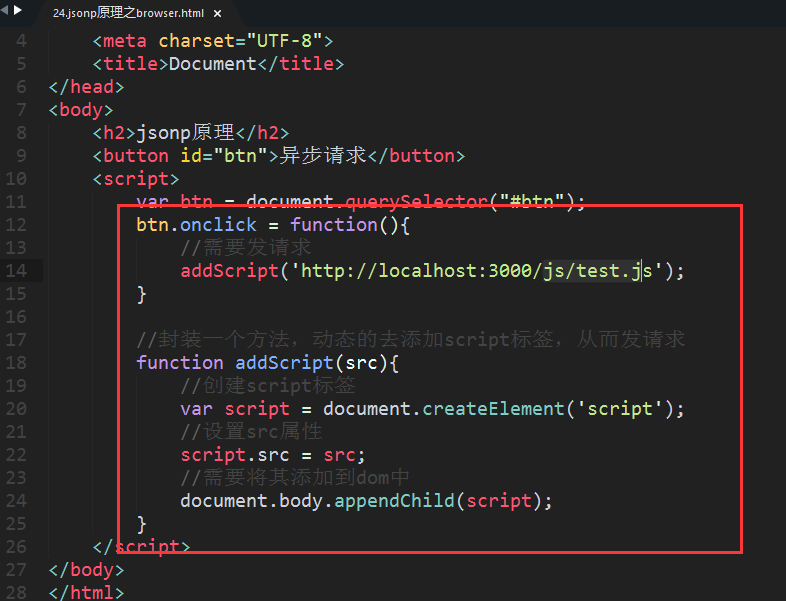
第一步，准备静态页面



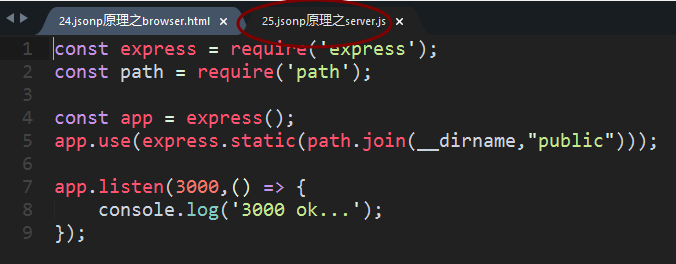
第二步，在点击button的时候，需要发起异步请求

不能使用xhr来http请求，也不使用jquery封装的ajax相关方法。

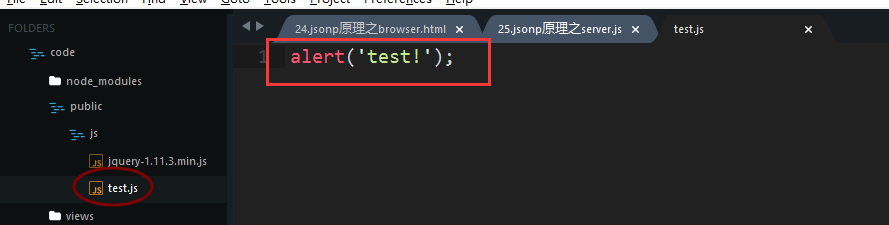
编写代码如下：



第三步，在3000端口提供http server并托管静态资源



第四步，创建test.js，如下：

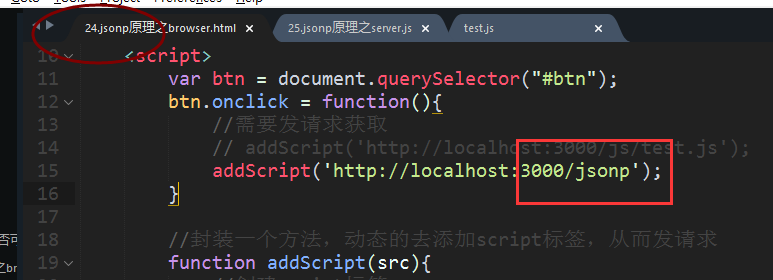


执行如下：



说明，这个请求是可以发出的。

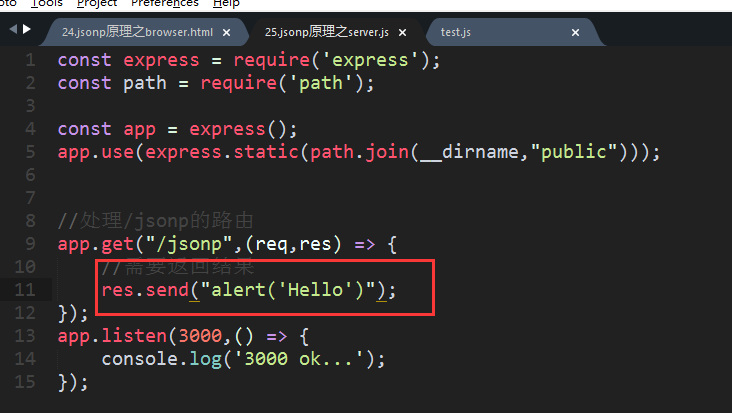
第五步，使用addScript发一个普通的请求



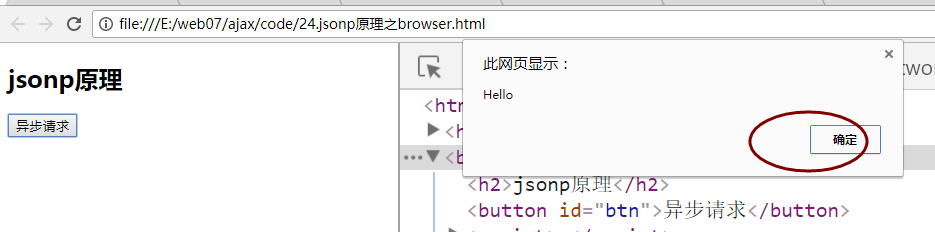
只要是script标签的src，写任何地址都可以。能够返回结果并执行，则是另外一回事。

第六步，在服务端，处理/jsonp请求

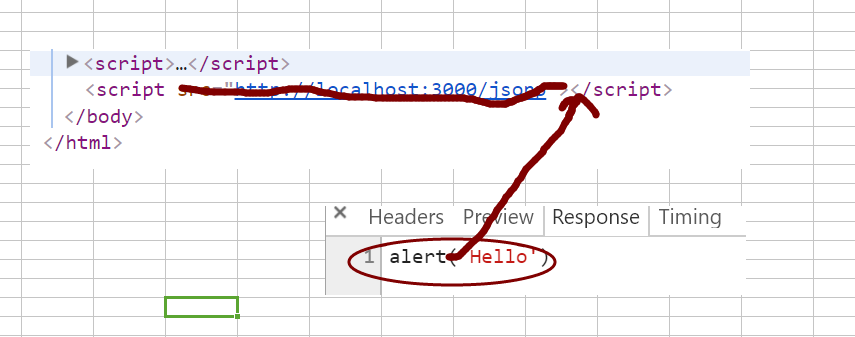
在返回的时候，返回一段js代码，如下：



重启，执行，如下：



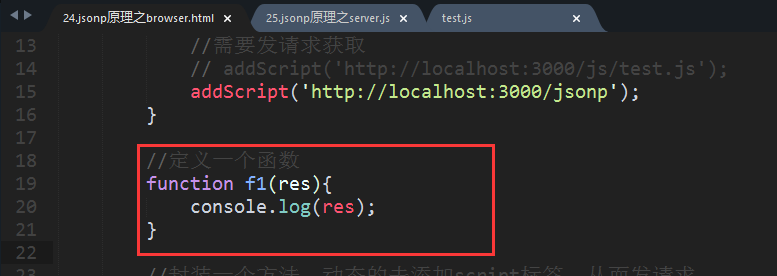
之所以，可以执行，那是因为，返回的是一段js代码，就好比是直接插入到如下位置：



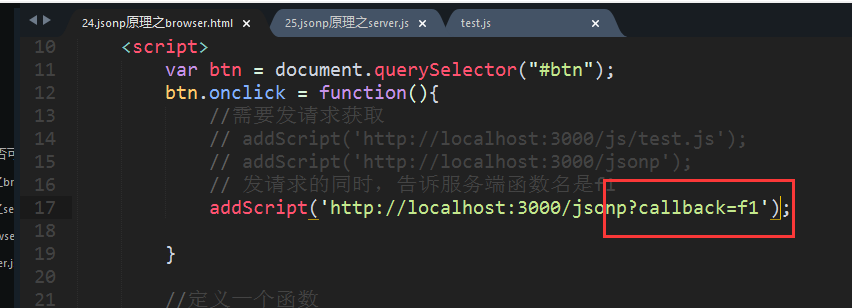
第七步，需要在服务端返回相应的数据，在浏览端获取并操作

目前我们是直接在服务端输出js代码，在浏览器端我们就不能拿到返回的结果，进行二次处理。换言之，浏览器端是被动的执行js代码，而不是获取返回的结果，主动的去操作。

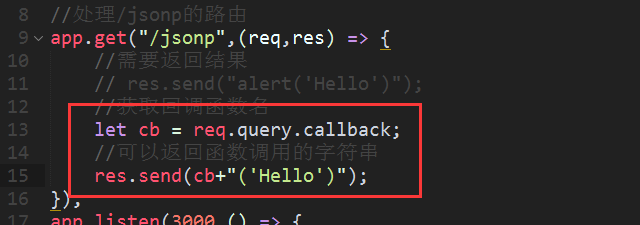
在浏览器定义一个函数，如下：



需要告诉服务端，我定义的函数叫什么名字？



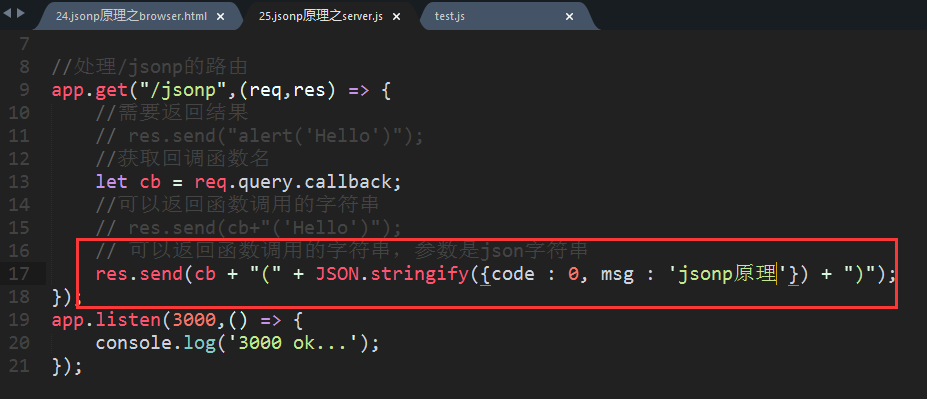
如此一来，就可以在服务端获取这个函数的名字，如下：



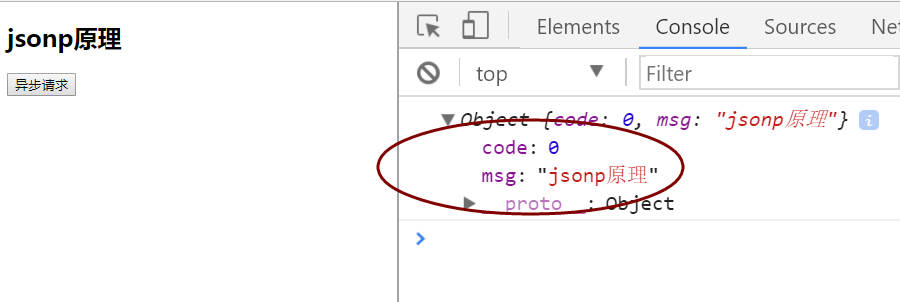
测试，如下：



实际上，在服务端返回的通常是json字符串，如下：



测试如下：



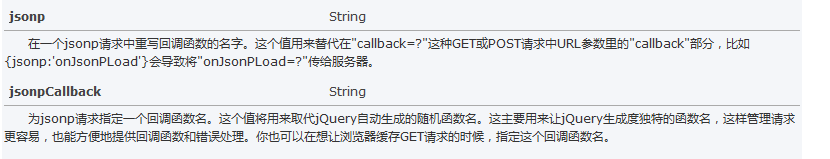
再次回顾jsonp的概念：

将json数据包装起来并返回的一种方式 ----- 协议

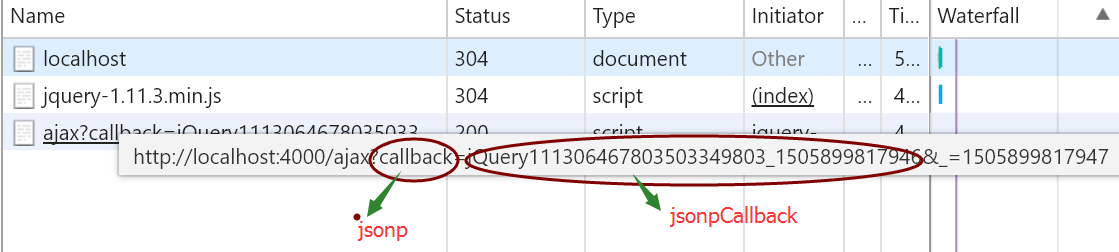
在使用jsonp实现跨域的时候，它是需要将json数据作为函数的参数（packing的过程），最终在服务端将包装好的字符串作为结果返回给浏览器端。

### jquery中jsonp相关设置

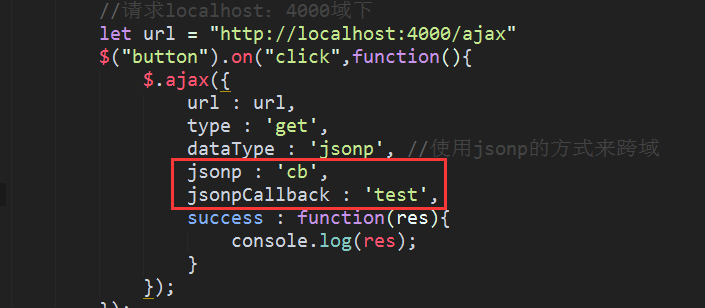
在jquery中的ajax方法中，和jsonp相关的还有如下两个设置：



默认如下：



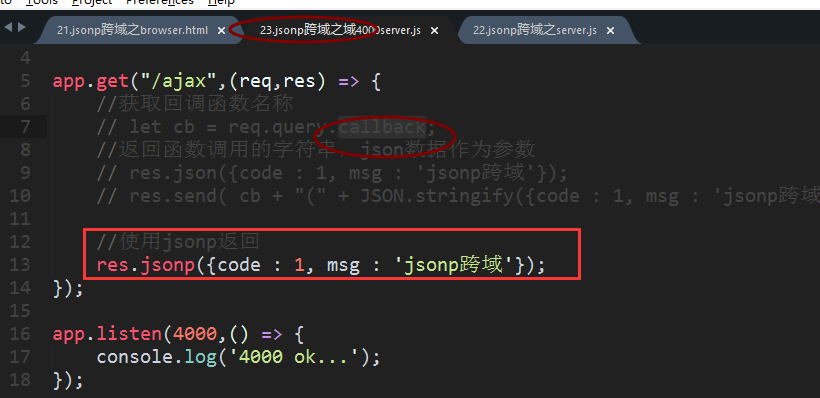
可以自己设置，如下：



测试如下：



但是，需要注意，res的jsonp方法，要求回调函数名必须是callback，否则出错，如下：



三种方式都已经讲解完毕。

在实际开发的时候，具体使用哪种方式，需要看情况来定。

作为前端，有时候是主动的，有时候是被动的。

而且，可以将cors和代理结合到一起，可以天下无敌。（大家自己写一写）