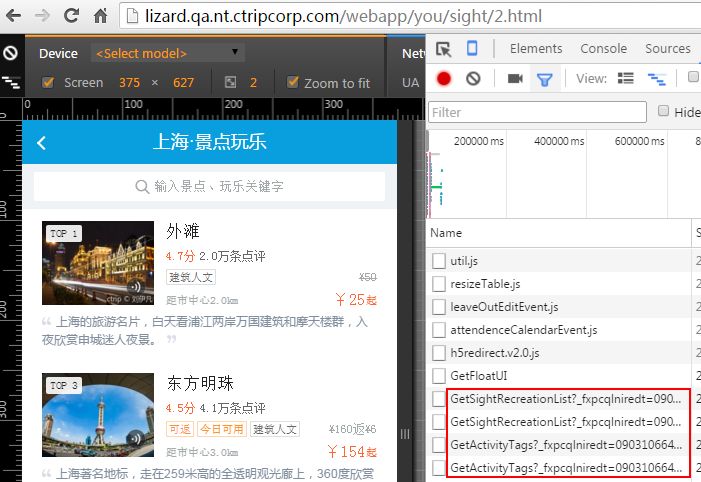
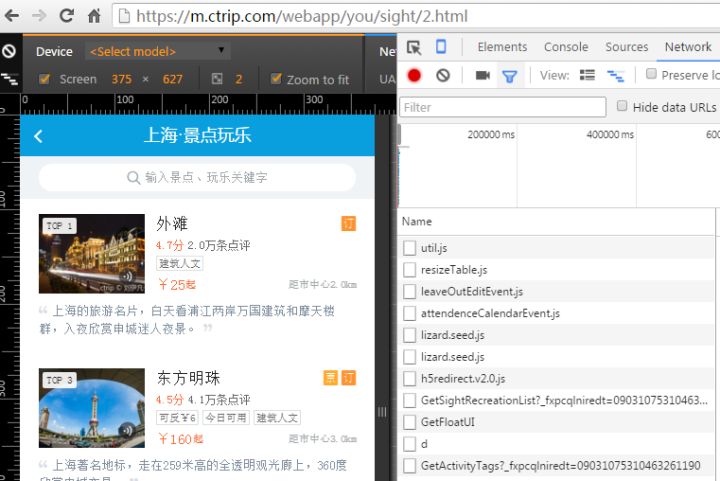
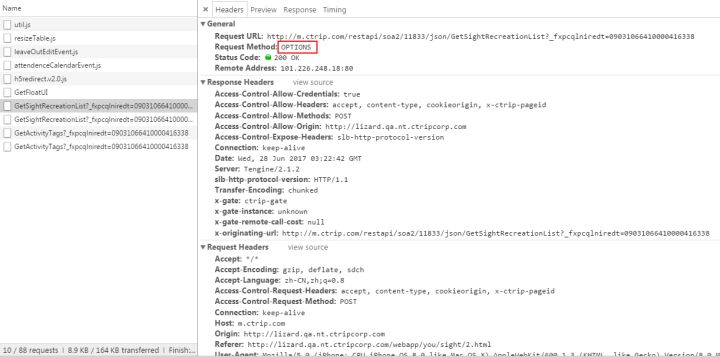
浏览器在跨域(或者port)请求中，在正式通信前会增加一次HTTP查询请求，称为"预检"请求（preflight）。

览器先询问服务器，当前网页所在的域名是否在服务器的许可名单之中，以及可以使用哪些HTTP动词和头信息字段。只有得到肯定答复，浏览器才会发出正式的XMLHttpRequest请求，否则就报错。

如上两张图片，图2就是本地环境，本地环境请求接口的时候，每次请求接口之前都会多一次OPTIONS请求。



CORS

CORS是一个W3C标准，全称是"跨域资源共享"（Cross-origin resource sharing）。

它允许浏览器向跨源服务器，发出[XMLHttpRequest](http://link.zhihu.com/?target=http%3A//www.ruanyifeng.com/blog/2012/09/xmlhttprequest_level_2.html)请求，从而克服了AJAX只能[同源](http://link.zhihu.com/?target=http%3A//www.ruanyifeng.com/blog/2016/04/same-origin-policy.html)使用的限制。

CORS需要浏览器和服务器同时支持。目前，所有浏览器都支持该功能，IE浏览器不能低于IE10。

整个CORS通信过程，都是浏览器自动完成，不需要用户参与。对于开发者来说，CORS通信与同源的AJAX通信没有差别，代码完全一样。浏览器一旦发现AJAX请求跨源，就会自动添加一些附加的头信息，有时还会多出一次附加的请求，但用户不会有感觉。

因此，实现CORS通信的关键是服务器。只要服务器实现了CORS接口，就可以跨源通信。

两种请求

浏览器将CORS请求分为两类：简单请求（simple request）和非简单请求（not-so-simple request）。

**同时满足**以下条件，就是简单请求：

（1) 请求方法是以下三种方法之一：

* HEAD
* GET
* POST

（2）HTTP的头信息不超出以下几种字段：

* Accept
* Accept-Language
* Content-Language
* Last-Event-ID
* Content-Type：只限于三个值application/x-www-form-urlencoded、multipart/form-data、text/plain

简单请求

对于简单请求，浏览器直接发出CORS请求。具体来说，就是在头信息之中，增加一个Origin字段。

Origin字段用来说明，本次请求来自哪个源（协议 + 域名 + 端口）。服务器根据这个值，决定是否同意这次请求。

如果Origin指定的源，不在许可范围内，服务器会返回一个正常的HTTP回应。浏览器发现，这个回应的头信息没有包含Access-Control-Allow-Origin字段（详见下文），就知道出错了，从而抛出一个错误，被XMLHttpRequest的onerror回调函数捕获。注意，这种错误无法通过状态码识别，因为HTTP回应的状态码有可能是200。

如果Origin指定的域名在许可范围内，服务器返回的响应，会多出几个头信息字段。都以**Access-Control-**开头：

**（1）Access-Control-Allow-Origin**

该字段是必须的。它的值要么是请求时Origin字段的值，要么是一个\*，表示接受任意域名的请求。

需要注意的是，如果要发送Cookie，Access-Control-Allow-Origin就不能设为星号，必须指定明确的、与请求网页一致的域名。同时，Cookie依然遵循同源政策，只有用服务器域名设置的Cookie才会上传，其他域名的Cookie并不会上传，且（跨源）原网页代码中的document.cookie也无法读取服务器域名下的Cookie。

**（2）Access-Control-Allow-Credentials**

该字段可选。它的值是一个布尔值，表示是否允许发送Cookie。默认情况下，Cookie不包括在CORS请求之中。设为true，即表示服务器明确许可，Cookie可以包含在请求中，一起发给服务器。这个值也只能设为true，如果服务器不要浏览器发送Cookie，删除该字段即可。

**（3）Access-Control-Expose-Headers**

该字段可选。CORS请求时，XMLHttpRequest对象的getResponseHeader()方法只能拿到6个基本字段：Cache-Control、Content-Language、Content-Type、Expires、Last-Modified、Pragma。如果想拿到其他字段，就必须在Access-Control-Expose-Headers里面指定。

非简单请求

非简单请求是那种对服务器有特殊要求的请求，比如请求方法是PUT或DELETE，或者Content-Type字段的类型是application/json。

我工作中写的所有页面拉的接口都是非简单请求。



非简单请求的CORS请求，会在正式通信之前，增加一次HTTP查询请求，称为"预检"请求（preflight）。

浏览器先询问服务器，当前网页所在的域名是否在服务器的许可名单之中，以及可以使用哪些HTTP动词和头信息字段。只有得到肯定答复，浏览器才会发出正式的XMLHttpRequest请求，否则就报错。

在页面域名与接口域名不一致的情况下，就出现了每次请求前先发送一个options请求的问题。

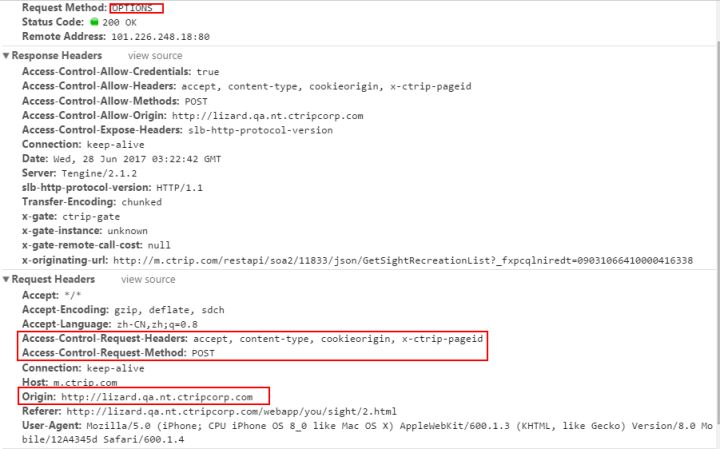
OPTIONS请求头信息中，除了Origin字段，还至少会多两个特殊字段：

**（1）Access-Control-Request-Method**

该字段是必须的，用来列出浏览器的CORS请求会用到哪些HTTP方法。

**（2）Access-Control-Request-Headers**

该字段是一个逗号分隔的字符串，指定浏览器CORS请求会额外发送的头信息字段。

至于其他乱七八糟的字段，现在的我还用不到也不懂，将会慢慢深入了解。

服务器收到预检请求后，检查了Origin、Access-Control-Request-Method和Access-Control-Request-Headers字段以后，确认允许跨源请求，就可以做出回应。

上面的HTTP回应中，关键的是Access-Control-Allow-Origin字段，表示[http://lizard.qa.nt.ctripcorp.com](http://link.zhihu.com/?target=http%3A//lizard.qa.nt.ctripcorp.com)可以请求数据。该字段也可以设为星号，表示同意任意跨源请求。

如果浏览器否定了"预检"请求，会返回一个正常的HTTP回应，但是没有任何CORS相关的头信息字段。这时，浏览器就会认定，服务器不同意预检请求，因此触发一个错误，被XMLHttpRequest对象的onerror回调函数捕获。控制台会打印出如下的报错信息。

XMLHttpRequest cannot load [http://lizard.qa.nt.ctripcorp.com](http://link.zhihu.com/?target=http%3A//lizard.qa.nt.ctripcorp.com)  
Origin [http://lizard.qa.nt.ctripcorp.com](http://link.zhihu.com/?target=http%3A//lizard.qa.nt.ctripcorp.com) is not allowed by Access-Control-Allow-Origin.

其他字段中**Access-Control-Max-Age**用来指定本次预检请求的有效期，单位为秒。该字段可选。

与JSONP的对比

CORS与JSONP的使用目的相同，但是比JSONP更强大。

JSONP只支持GET请求，JSONP的优势在于支持老旧浏览器。