自：<http://www.ruanyifeng.com/blog/2016/04/cors.html>

# 跨域资源共享 CORS 详解

作者： [阮一峰](http://www.ruanyifeng.com/)

日期： [2016年4月12日](http://www.ruanyifeng.com/blog/2016/04/)

[](http://www.zhufengpeixun.cn/main/index.html?ref=ruanyifeng)

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://www.ruanyifeng.com/blog/2012/09/xmlhttprequest_level_2.html)请求，从而克服了AJAX只能[同源](http://www.ruanyifeng.com/blog/2016/04/same-origin-policy.html)使用的限制。

本文详细介绍CORS的内部机制。



（图片说明：摄于阿联酋艾因（Al Ain）的绿洲公园）

## 一、简介

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。

## 二、两种请求

浏览器将CORS请求分成两类：简单请求（simple request）和非简单请求（not-so-simple request）。

只要同时满足以下两大条件，就属于简单请求。

（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

凡是不同时满足上面两个条件，就属于非简单请求。

浏览器对这两种请求的处理，是不一样的。

## 三、简单请求

### 3.1 基本流程

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。

下面是一个例子，浏览器发现这次跨源AJAX请求是简单请求，就自动在头信息之中，添加一个Origin字段。

GET /cors HTTP/1.1

Origin: [http://api.bob.com](http://api.bob.com/)

Host: api.alice.com

Accept-Language: en-US

Connection: keep-alive

User-Agent: Mozilla/5.0...

上面的头信息中，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。

Access-Control-Allow-Origin: [http://api.bob.com](http://api.bob.com/)

Access-Control-Allow-Credentials: true

Access-Control-Expose-Headers: FooBar

Content-Type: text/html; charset=utf-8

上面的头信息之中，有三个与CORS请求相关的字段，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。

**（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。上面的例子指定，getResponseHeader('FooBar')可以返回FooBar字段的值。

### 3.2 withCredentials 属性

上面说到，CORS请求默认不发送Cookie和HTTP认证信息。如果要把Cookie发到服务器，一方面要服务器同意，指定Access-Control-Allow-Credentials字段。

Access-Control-Allow-Credentials: true

另一方面，开发者必须在AJAX请求中打开withCredentials属性。

var xhr = new XMLHttpRequest();

xhr.withCredentials = true;

否则，即使服务器同意发送Cookie，浏览器也不会发送。或者，服务器要求设置Cookie，浏览器也不会处理。

但是，如果省略withCredentials设置，有的浏览器还是会一起发送Cookie。这时，可以显式关闭withCredentials。

xhr.withCredentials = false;

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。

## 四、非简单请求

### 4.1 预检请求

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。

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。

下面是一段浏览器的JavaScript脚本。

var url = '<http://api.alice.com/cors>';

var xhr = new XMLHttpRequest();

xhr.open('PUT', url, true);

xhr.setRequestHeader('X-Custom-Header', 'value');

xhr.send();

上面代码中，HTTP请求的方法是PUT，并且发送一个自定义头信息X-Custom-Header。

浏览器发现，这是一个非简单请求，就自动发出一个"预检"请求，要求服务器确认可以这样请求。下面是这个"预检"请求的HTTP头信息。

OPTIONS /cors HTTP/1.1

Origin: [http://api.bob.com](http://api.bob.com/)

Access-Control-Request-Method: PUT

Access-Control-Request-Headers: X-Custom-Header

Host: api.alice.com

Accept-Language: en-US

Connection: keep-alive

User-Agent: Mozilla/5.0...

"预检"请求用的请求方法是OPTIONS，表示这个请求是用来询问的。头信息里面，关键字段是Origin，表示请求来自哪个源。

除了Origin字段，"预检"请求的头信息包括两个特殊字段。

**（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，上例是PUT。

**（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，上例是X-Custom-Header。

### 4.2 预检请求的回应

服务器收到"预检"请求以后，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。

HTTP/1.1 200 OK

Date: Mon, 01 Dec 2008 01:15:39 GMT

Server: Apache/2.0.61 (Unix)

Access-Control-Allow-Origin: [http://api.bob.com](http://api.bob.com/)

Access-Control-Allow-Methods: GET, POST, PUT

Access-Control-Allow-Headers: X-Custom-Header

Content-Type: text/html; charset=utf-8

Content-Encoding: gzip

Content-Length: 0

Keep-Alive: timeout=2, max=100

Connection: Keep-Alive

Content-Type: text/plain

上面的HTTP回应中，关键的是Access-Control-Allow-Origin字段，表示http://api.bob.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。

Access-Control-Allow-Origin: \*

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://api.alice.com.

Origin http://api.bob.com is not allowed by Access-Control-Allow-Origin.

服务器回应的其他CORS相关字段如下。

Access-Control-Allow-Methods: GET, POST, PUT

Access-Control-Allow-Headers: X-Custom-Header

Access-Control-Allow-Credentials: true

Access-Control-Max-Age: 1728000

**（1）Access-Control-Allow-Methods**

该字段必需，它的值是逗号分隔的一个字符串，表明服务器支持的所有跨域请求的方法。注意，返回的是所有支持的方法，而不单是浏览器请求的那个方法。这是为了避免多次"预检"请求。

**（2）Access-Control-Allow-Headers**

如果浏览器请求包括Access-Control-Request-Headers字段，则Access-Control-Allow-Headers字段是必需的。它也是一个逗号分隔的字符串，表明服务器支持的所有头信息字段，不限于浏览器在"预检"中请求的字段。

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

该字段与简单请求时的含义相同。

**（4）Access-Control-Max-Age**

该字段可选，用来指定本次预检请求的有效期，单位为秒。上面结果中，有效期是20天（1728000秒），即允许缓存该条回应1728000秒（即20天），在此期间，不用发出另一条预检请求。

### 4.3 浏览器的正常请求和回应

一旦服务器通过了"预检"请求，以后每次浏览器正常的CORS请求，就都跟简单请求一样，会有一个Origin头信息字段。服务器的回应，也都会有一个Access-Control-Allow-Origin头信息字段。

下面是"预检"请求之后，浏览器的正常CORS请求。

PUT /cors HTTP/1.1

Origin: [http://api.bob.com](http://api.bob.com/)

Host: api.alice.com

X-Custom-Header: value

Accept-Language: en-US

Connection: keep-alive

User-Agent: Mozilla/5.0...

上面头信息的Origin字段是浏览器自动添加的。

下面是服务器正常的回应。

Access-Control-Allow-Origin: [http://api.bob.com](http://api.bob.com/)

Content-Type: text/html; charset=utf-8

上面头信息中，Access-Control-Allow-Origin字段是每次回应都必定包含的。

## 五、与JSONP的比较

CORS与JSONP的使用目的相同，但是比JSONP更强大。

JSONP只支持GET请求，CORS支持所有类型的HTTP请求。JSONP的优势在于支持老式浏览器，以及可以向不支持CORS的网站请求数据。

（完）

### 文档信息

* 版权声明：自由转载-非商用-非衍生-保持署名（[创意共享3.0许可证](http://creativecommons.org/licenses/by-nc-nd/3.0/deed.zh)）
* 发表日期： 2016年4月12日
* 更多内容： [档案](http://www.ruanyifeng.com/blog/archives.html) » [JavaScript](http://www.ruanyifeng.com/blog/javascript/)
* 文集：[《前方的路》](http://www.ruanyifeng.com/road)，[《未来世界的幸存者》](http://www.ruanyifeng.com/survivor)
* 社交媒体：[twitter](https://twitter.com/ruanyf)，[weibo](http://weibo.com/ruanyf)

[](https://2018.qconshanghai.com/?utm_source=ryf&utm_campaign=9&utm_medium=banner&utm_term=0815)

[](https://ke.qq.com/next/index.html?from=175763)

## 相关文章

* **2018.07.08:**[Web Worker 使用教程](http://www.ruanyifeng.com/blog/2018/07/web-worker.html)

一、概述

* **2018.07.04:**[浏览器数据库 IndexedDB 入门教程](http://www.ruanyifeng.com/blog/2018/07/indexeddb.html)

一、概述

* **2018.06.18:**[JavaScript 的 this 原理](http://www.ruanyifeng.com/blog/2018/06/javascript-this.html)

一、问题的由来 学懂 JavaScript 语言，一个标志就是理解下面两种写法，可能有不一样的结果。

* **2018.03.20:**[Node 调试工具入门教程](http://www.ruanyifeng.com/blog/2018/03/node-debugger.html)

JavaScript 程序越来越复杂，调试工具的重要性日益凸显。客户端脚本有浏览器，Node 脚本怎么调试呢？

## 广告[（购买广告位）](http://www.ruanyifeng.com/support.html)

[](https://cn.udacity.com/course/ai-for-trading--nd880/?utm_source=ruanyifeng_b&utm_medium=banner&utm_campaign=aitnd)

[](https://www.apicloud.com/video_list?uzchannel=41)

## 留言（104条）

**Se7en 说：**

阮老师讲得非常全面透彻，看完这篇文章就会对浏览器的同源策略和跨域请求了解清楚了，堪称经典。

2016年4月12日 07:11 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355561) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**杨铮 说：**

asp.net web api 中的坑。  
web.config 中 remove name="OPTIONSVerbHandler" 需要去掉。

2016年4月12日 08:38 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355563) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**光君**](http://www.graysheep.me/)**说：**

可以可以，老师这篇文章很关键。但是想多问一句CORS在哪些场景下应用比较多？

2016年4月12日 09:33 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355567) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**Lxxyx**](http://lxxyx.win/)**说：**

正在最近在纠结跨域请求的相关问题，而已有资料的质量良莠不齐，阮一峰老师的两篇博文真的超级赞。

2016年4月12日 09:55 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355568) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**Junyi 说：**

Access-Control-Allow-Origin支持多域名：http://junyi.me/blog/s17.html

2016年4月12日 19:44 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355591) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**四喜 说：**

文中提到“CORS请求默认不发送Cookie”，通过服务端响应 Access-Control-Allow-Credentials: true 来控制。但如果接口依赖Cookie来完成一些处理（比如登录态），那第一次发送CORS请求“默认不发送Cookie”的话，服务端接受不到Cookie，岂不会有问题？

从实验上来看，Access-Control-Allow-Credentials: true 好像并没有控制Cookie是否发送，因为即使之前没有任何CORS请求（自然也没有往客户端发送该Header），第一次CORS请求也会带上所有该域能访问的Cookie。

测试环境：Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_11\_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.110 Safari/537.36

2016年4月13日 10:17 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355606) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**阮一峰 说：**

@四喜，

我试了一下，如果设为 withCredentials = false ，就不会发送 Cookie 了。如果省略这一行，确实会发送。但 MDN 明确说，默认值是 false。https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/withCredentials

我原来的说法表达得不确切，我改一下。

2016年4月13日 11:19 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355609) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**waylau**](http://waylau.com/)**说：**

Access-Control-Allow-Origin 字段，如何表示允许一个以上的网站访问(不是\*这种全部)，比如 允许 [http://api.bob.com、](http://api.bob.xn--com-6y3b/) [http://api.alice.com](http://api.alice.com/)两个网站。

Access-Control-Allow-Origin 字段能实现吗？我这边试下，好像不行。

2016年4月13日 16:50 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355628) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**mrsu**](http://mrsu.me/)**说：**

简单请求的 Content-Type 包含 text/plain 而不是 text/html 感觉不科学啊～

2016年4月13日 19:53 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355634) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**OMEGA 说：**

简单请求包含connection首部，但是文中指出的不符？

（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

2016年4月15日 11:49 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355684) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**张歆琳 说：**

4.2节里有笔误Access-Control-Allow-Headers: true 应该是 Access-Control-Allow-Credentials: true

2016年4月16日 00:00 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355690) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**阮一峰 说：**

@张歆琳：

谢谢指出，已经改正了。

2016年4月16日 11:32 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355693) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**skyer 说：**

如果浏览器否定了"预检"请求，  
----  
这个应该是『服务器否定了"预检"请求』吧。

2016年4月18日 18:41 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355736) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**迹忆**](http://www.onmpw.com/)**说：**

写的非常好，感谢阮老师的分享

2016年4月19日 13:29 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355746) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**文刀 说：**

文章写的非常好，又掌握了一个重要的知识点！感谢阮兄分享

2016年4月20日 11:54 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355767) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**Sia 说：**

有一个问题想请教一下，在淘宝登陆后，打开天猫，会发现也是已登录状态，这是怎么做的呢？

2016年4月24日 10:36 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355858) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**GlacJAY**](https://glacjay.info/)**说：**

4.2 节里面，「服务器回应的其他CORS相关字段如下」中，有个字段写错了吧？写了两次「Allow-Headers」。

2016年4月26日 10:05 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355882) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**大头 说：**

看阮老师一篇文章，胜读十本书

2016年4月26日 14:57 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-355891) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**二点零 说：**

学习了，两篇文章讲得非常详细，感谢

2016年5月 5日 15:07 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-356078) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**小小的大二学生 说：**

还有IE的ActiveXObject

2016年5月 8日 21:38 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-356124) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**纪广财**](http://shanshuidan.com/)**说：**

受教，谢谢阮老师~

2016年5月24日 22:40 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-356593) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**如理 说：**

**引用大头的发言：**

看阮老师一篇文章，胜读十本书

你看书只看序的吧？

2016年5月26日 17:32 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-356653) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**echo 说：**

**引用光君的发言：**

可以可以，老师这篇文章很关键。但是想多问一句CORS在哪些场景下应用比较多？

restful api

2016年6月 3日 17:59 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-357085) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**quay 说：**

**引用echo的发言：**

restful api

有些疑问：  
1）为啥看到一些公开的API貌似没有origin   
2）restful api 、 web api 、cors 、web service 目前流行哪个做API呢   
3） cors IE10才可能，兼容性如何解决呢？

2016年6月11日 22:45 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-357479) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**cmxz 说：**

关于Credentials，重要的不是会不会发送，而是cookie发送到服务端后如果response header中无Access-Control-Allow-Credentials: true，那么浏览器会报异常，js无法拿到跨域请求的response对象

2016年6月19日 00:45 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-359398) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**甘训奏**](http://ganxunzou.com/)**说：**

我的请求：  
```  
$.ajax({  
type: "POST",  
url: "http://192.168.0.103:8081/forum/servlet/HelloServlet",  
//contentType:"application/json",  
data: {uName:$("#uName").val(), uPwd:$("#uPwd").val()},  
dataType: "json",  
success: function(data){  
$('#resText').empty(); // 清空resText里面的所有内容  
var html = '';   
$.each(data, function(commentIndex, comment){  
html += '<div class="comment"><h6>' + comment['username']  
+ ':</h6>

+ '</p></div>';  
});  
$('#resText').html(html);  
}  
});  
```

服务端接收到后增加CORS处理。  
```  
response.setContentType("text/html; charset=utf-8");  
response.addHeader("Access-Control-Allow-Origin", "\*");  
response.addHeader("Access-Control-Allow-Methods", "\*");  
response.addHeader("Access-Control-Max-Age", "100");  
response.addHeader("Access-Control-Allow-Headers", "X-Custom-Header,accept, content-type");  
response.addHeader("Access-Control-Allow-Credentials", "false");  
```

2016年6月22日 13:09 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-359596) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**甘训奏**](http://ganxunzou.com/)**说：**

===========================================  
为什么留言不能超过1200个字，接上一个提问  
===========================================

不加 `contentType="application/json"`正常访问后台。增加`contentType="application/json"`后提示：  
```  
XMLHttpRequest cannot load <http://192.168.0.103:8081/forum/servlet/HelloServlet.> Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://127.0.0.1:8020' is therefore not allowed access.  
```  
从错误看，应该是preflight （预检）请求涉及到跨域问题了。现在问题是preflight 的请求响应设置在哪边处理呢？

2016年6月22日 13:11 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-359597) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**潘逸飞 说：**

写的超赞！

2016年6月29日 17:03 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-359899) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**赵帅 说：**

阮老师想问你个问题，我目前也是在Cors跨域这块遇到了难题，1.我在复杂请求时，PUT，首先预检通过，可以看到status code:200状态码，2.然后第二次要做真实的修改操作，直接报500状态码，服务器内部错误，响应头里面没有Origin,Headers,Method这三项.请问这是什么情况呢？烦请阮老师帮忙看下。不胜感谢！

2016年6月30日 15:16 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-359931) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**eviltear 说：**

想问阮老师，跨协议请求有类似的解决方案吗？如http请求https，谢谢。

2016年7月25日 10:53 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-361155) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**Silas 说：**

提一点：  
做auth时，xhr.setRequestHeader，key 不能是 token 字符串。

2016年8月22日 11:09 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-362889) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**Slimmer 说：**

**引用光君的发言：**

可以可以，老师这篇文章很关键。但是想多问一句CORS在哪些场景下应用比较多？

我们项目目前使用Restful架构，由于前后不在台服务器，处处需跨源

2016年9月 2日 14:05 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-363624) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**xylsh 说：**

**引用Slimmer的发言：**

我们项目目前使用Restful架构，由于前后不在台服务器，处处需跨源

我在想，即使不在同一台服务器，如果前台用nginx做请求转发，把不同请求转发到不同服务器，这样，从浏览器的角度来看，域名就没有发生变化，也就不用跨域。我的想法对吗？

2016年9月27日 17:42 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-365284) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**cmxz 说：**

Access-Control-Max-Age字段在webkit的浏览器上默认最大时间为5分钟，若设置超过5分钟则不会生效，代码见https://cs.chromium.org/chromium/src/third\_party/WebKit/Source/core/loader/CrossOriginPreflightResultCache.cpp?rcl=1399481969&l=44  
另外，chrome实际上有bug，即使Access-Control-Max-Age设置为600，下次请求时依然会发送options请求，此bug在2012年被提出，但至今未修复。具体见 <https://bugs.chromium.org/p/chromium/issues/detail?id=131368>

2016年10月 8日 11:45 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-365885) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**cmxz 说：**

可以在http://test-cors.org/ 测试

2016年10月 8日 11:46 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-365887) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**cmxz 说：**

我之前的表述可能有问题。

经过测试，chrome现阶段（54.0.2840.50 beta-m (64-bit)）是支持对options的请求缓存5分钟的

我之前之所以会觉得没有缓存，是因为我原以为preflight-result-cache是以域名为单位的cache，但看了https://www.w3.org/TR/cors/#preflight-result-cache 后发现cache是以origin+url+credentials+method+header在max-age内cache的，所以不同url（get参数不同也算的）是会发送多次options请求的

2016年10月 9日 15:08 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-365976) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**第六天魔王**](http://www.njwangchuan.com/)**说：**

今天正好我女朋友面试，被问到这个问题，我觉得即便是研究生想说透也不是件很容易的事情。

我本人实际项目中倒是遇到过跨域问题，一个业务场景是用户上传图片然后切图后再保存，上传图片到第三方存储后（我用的是aws的s3+cloudfront），前端js类库从cdn获取原图时用的是ajax方法。在s3设置cors策略，在cloudfront设置http预检策略，结合阮老师的文章，基本理解到位了，非常感谢！

2016年10月10日 14:49 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-366050) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**李超 说：**

在3.1节中您讲到：

> 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。

请问 许可范围 是在什么地方指定的呢？谢谢！

2016年10月16日 10:55 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-366448) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**billow 说：**

目前为止看到的最深入浅出的解释！

2016年10月20日 17:44 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-366750) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**久保与二弦琴 说：**

您好，在测试这个问题的时候，发现即便服务器端不设置Access-Control-Allow-Origin头，服务器依然会返回数据，可以用fiddler抓取到，和设置头信息后的区别只是浏览器页面不显示数据而已。能否表明，CORS设置与否只是影响浏览器是否显示服务器返回的数据，而无法阻止服务器要不要返回数据?

2016年11月 5日 13:24 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-367818) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**严泽玮 说：**

ie8通过XDomainRequest能支持CORS，文中说要不低于ie10，难道我理解错了。

2016年11月17日 17:42 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-368637) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**赵启发 说：**

“上面说到，CORS请求默认不发送Cookie和HTTP认证信息。如果要把Cookie发到服务器，一方面要服务器同意，指定Access-Control-Allow-Credentials字段。”

对于简单请求，其并没有预检请求。也就是说收到服务器响应的时候，简单请求已经发出去了，也就是说简单请求发送的时候不管是带不带cookie，这个决策已经完成了。那么这时候server返回Access-Control-Allow-Credentials字段的意义是什么呢？

1. 告诉浏览器其实服务器并没有处理你发送过来的cookie，假设浏览器发了cookie并期待cookie生效以达成某种目的的话，用来帮助定位问题？  
2. 告诉浏览器这个信息，下次再发同样的跨域请求的时候就知道该不该发送cookie了？

2016年11月17日 20:52 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-368652) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**cxl253386709 说：**

最近要把web端已有的应用扩展到移动端h5轻应用上，遇到了跨域访问一个诡异的问题。在服务端设置好Access-Control-Allow-Origin：\* ；Access-Control-Allow-Credentials： "true" 后，两个系统做了cas，然后开发环境一切ok。等到项目准备上线时，打包进weblogic，居然发现访问失效。具体表现是当请求的结果超出一定数据量以后（因为可以指定每页条数，所以可以一点点的加返回数据量，当到了一个临界值以后，突然出现问题，比如30条的时候没事，31条突然就没数据了），从浏览器观察返回的response为空。在weblogic开了debug模式，发现后端一切正常没有任何错误，也确实查了数据库也确实返回了。后来为了验证这个问题，把同一个包又放到tomcat上跑，发现能返回数据量的临界值更小了，同一个包放在本地jetty上就没这个问题。所以我怀疑是不是各个应用服务器在处理跨域返回数据量时有一个默认值，可以配置的？我找了文档却没发现有这个设置。百思不得其解，卡了我好几个周了，求大神解惑！！！  
今天我又做了几个实验，这个限额大约是20k的样子！！

2016年11月30日 14:36 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-369470) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**lili 说：**

大神，有没有cros跨域请求的具体实例呀？我在使用cros跨域上传文件过程中从后台返回的数据前台接收不到，status是0，急用，麻烦您了

2016年12月13日 17:35 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-370334) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**DemoorBug 说：**

**引用Silas的发言：**

提一点：  
做auth时，xhr.setRequestHeader，key 不能是 token 字符串。

我就遇到了这种问题，angular 拦截器设置了 req.headers['token'] = ‘’； 跨域就无法通过预检测器，请问有什么方法么，我也设置了  
res.setHeader('Access-Control-Allow-Headers','token');  
但还是无法实现

2016年12月22日 15:43 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-370835) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**夏笛 说：**

cors并不是所有浏览器都支持吧，opera mini就是一个例子，现在做这套策略还应该考虑降级问题

2016年12月30日 14:50 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-371339) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**arkin**](http://www.codesec.net/view/154574.html)**说：**

Access-Control-Max-Age 在chrome控制台注意有disabled cache的选项,  
不然会被坑,我调了很久也不生效,信好网上看到.提醒一下大家.

2017年1月 4日 17:54 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-371606) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**和珅 说：**

Access-Control-Request-Method 这个方法是告诉服务端我将要用什么方法请求数据，而不是方法的罗列。  
补充withCredentials = true; 设置Access-Control-Allow-Credentials = true的同时Access-Control-Allow-Origin的值不能为“\*”，必须设置一个完整的域，只允许一条。

2017年1月 9日 10:02 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-371805) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**bsc2xp 说：**

**引用cmxz的发言：**

关于Credentials，重要的不是会不会发送，而是cookie发送到服务端后如果response header中无Access-Control-Allow-Credentials: true，那么浏览器会报异常，js无法拿到跨域请求的response对象

这么解释看得才明白，文章里说的没理解

2017年2月 9日 09:21 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-372585) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**leaf 说：**

**引用quay的发言：**

有些疑问：  
1）为啥看到一些公开的API貌似没有origin  
2）restful api、 web api 、cors 、web service 目前流行哪个做API呢  
3） cors IE10才可能，兼容性如何解决呢？

1）应该是那些应用有一个后台服务器，后台服务器调用的API  
2）restful api 是 web api 中的一种吧   
cors 只要浏览器 xhr 跨域请求都要面对的问题，和什么形式 api 无关   
web service 是很早之前的技术，现在应该是不流行了   
3）JSONP

2017年2月21日 16:31 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-373225) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**茉莉 说：**

**引用Sia的发言：**

有一个问题想请教一下，在淘宝登陆后，打开天猫，会发现也是已登录状态，这是怎么做的呢？

自己感觉是用同一个数据库吧 起码是用户表 订单表 之类的 通用的会有关联

2017年2月23日 18:19 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-373347) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**刘欣（Lx15）**](http://lx15.github.io/)**说：**

阮老师您好，我测试了一下。。。。在前后端分离的情况下，前端发put的非简单请求，并不能够跨源，这是为什么呢？ 后端采用node 接收请求。   
在前后端不分离的情况下，前端发put的非简单请求，是可以跨源的。

2017年2月24日 11:43 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-373357) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**吴宏伟 说：**

**引用杨铮的发言：**

asp.net web api 中的坑。  
web.config 中 remove name="OPTIONSVerbHandler" 需要去掉。

还真碰到这个坑了。

2017年2月24日 19:58 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-373377) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**Leesson**](https://github.com/Leesson)**说：**

感谢，介绍的很详细。  
另外请问，如果返回头里包含两个Access-Control-Allow-Origin符合规范吗？比如返回头同时包含如下两个：  
Access-Control-Allow-Origin：www.origin.com //(请求发出域名)  
Access-Control-Allow-Origin：\*

在实际中遇上过上述问题，后台代码和IIS分别为返回头添加了Access-Control-Allow-Origin。当请求发出域名和接收域名（后台）是同一域名时可正常得到结果，但不是统一域名时就会报跨域错误。  
阮老师能帮忙解释一下吗？谢谢

2017年2月27日 00:23 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-373470) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**Ryan 说：**

看文章的说法，浏览器决定是否跨域，那么服务端呢？简单的跨域，浏览器带着cookie请求服务器，服务器设置false，则不接收cookie？事实上cookie已经加载到request里了。  
第二个问题，简单的跨域中，如果服务端部允许跨域，那么服务端是直接返回空还是按照正常的逻辑返回了数据，只是到浏览器的时候被浏览器给拦截了？

2017年3月30日 23:31 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-375699) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**vic**](http://www.yxsss.com/)**说：**

@Ryan：

肯定啊 都是浏览器，如果不是浏览器 就不存在这个问题 比如服务器端程序 想请求谁就请求谁 ，

2017年3月31日 11:16 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-375717) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**ouyang 说：**

预检结果  
Response Headers  
Allow:GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, PATCH  
Content-Length:0  
Date:Fri, 31 Mar 2017 06:12:56 GMT  
Request Headers  
Accept:\*/\*  
Accept-Encoding:gzip, deflate, sdch  
Accept-Language:zh-CN,zh;q=0.8  
Access-Control-Request-Headers:authorization  
Access-Control-Request-Method:POST  
Connection:keep-alive  
Host:10.220.17.121:8080  
Origin:http://10.220.18.113:8020  
Referer:http://10.220.18.113:8020/xb/wx/201703013/login.html  
User-Agent:Mozilla/5.0 (Linux; Android 5.0; SM-G900P Build/LRX21T) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Mobile Safari/537.36  
如何让浏览器通过Origin呢 怎么设置？

2017年3月31日 14:20 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-375727) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**龙顺 说：**

我只能说，相见恨晚呐！看了几篇阮老师的博客下来，发现内容讲解都非常细致，很难找到比阮老师的更适合初学者的教程了。

关于JSONP，“JSONP的优势在于支持老式浏览器，以及可以向不支持CORS的网站请求数据”。支持老式浏览器是确实是其优势，但像不支持CORS的网站请求数据，我觉得就不是它的优势了，因为后端代码还是得针对JSONP做出相应的调整（返回json字符串的部分需要用传过去的回调函数名包起来），也很麻烦。反而CORS的方式，只需要在后端代码的过滤器上加上相应的响应头就行了，来得更方便。

2017年4月 7日 00:19 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-375973) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**gorden 说：**

阮老师，感谢您的文章。现在有一些不明白的地方希望跟你请教一下：  
服务器对于预检请求和实际请求是需要用不同的方法去拦截吗？还是这两个请求都是由同一个方法去处理的

2017年4月12日 23:34 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-376224) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**赵启发**](http://www.708luo.com/)**说：**

<https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Access_control_CORS>

这里找到答案了：  
“因为这是一个简单 GET 请求，所以浏览器不会对其发起“预检请求”。但是，如果服务器端的响应中未携带 Access-Control-Allow-Credentials: true ，浏览器将不会把响应内容返回给请求的发送者。”

2017年5月 3日 16:16 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-376850) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**j.zong**](http://jackzong.github.io/)**说：**

阮老师，3.1节中，【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】，这句话有误吧？如果Origin源不在指定的许可范围内，回应的头信息也会包含Access-Control-Allow-Origin字段吧，只是它的值与Origin源不匹配，就会执行onerror错误事件驱动程序

2017年5月 3日 21:18 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-376856) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**龙海洋 说：**

一个post跨域请求在“预检”请求OPTIONS中Origin为什么会为null，并且会报错：

Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'null' is therefore not allowed access. The response had HTTP status code 401.

OPTIONS请求http头部信息如下：

Accept:\*/\*  
Accept-Encoding:gzip, deflate, sdch, br  
Accept-Language:zh-CN,zh;q=0.8  
Access-Control-Request-Headers:authorization  
Access-Control-Request-Method:POST  
Connection:keep-alive  
Host:chinapopin.com  
Origin:null  
User-Agent:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.96 Safari/537.36

2017年5月10日 10:08 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-377004) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**syd\_2011 说：**

**引用严泽玮的发言：**

ie8通过XDomainRequest能支持CORS，文中说要不低于ie10，难道我理解错了。

xDomainRequest 只支持post,get 不给带cookie等等 大多数情况下等于没什么用

2017年5月16日 19:11 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-377208) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**阿豪 说：**

厉害了 阮老师，收获很多

2017年6月 1日 09:48 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-377611) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**阿辉 说：**

阮老师，这样看来，跨域始终都需要服务端支持才能实现对吗？ 并不只是纯粹前端设置就能成功的吗？

2017年6月 3日 17:54 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-377651) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**夜的眼睛 说：**

真心厉害，还是老师讲的简单易懂，一点就通！

2017年6月 7日 23:34 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-377749) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**贾越 说：**

没看懂 谁来教下我 前端和node 怎么设置？

2017年6月13日 13:35 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-377915) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**谷雨**](https://github.com/dsgdtc)**说：**

**引用茉莉的发言：**

自己感觉是用同一个数据库吧 起码是用户表 订单表 之类的 通用的会有关联

应该是通过CAS做的单点登录认证（SSO）

2017年6月23日 11:34 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-378147) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**daoSs 说：**

怎么有这么好的文章，结合工作中的实践，让我对CORS有了非常清楚的认识。

2017年6月28日 12:06 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-378262) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**ifu25**](http://www.vcoo.me/)**说：**

**引用杨铮的发言：**

asp.net web api 中的坑。  
web.config 中 remove name="OPTIONSVerbHandler" 需要去掉。

这真是绝世大坑！我搞了两天，查了无数资料，竟然有那么多网站说要把OPTIONSVerbHandler和WebDAV删除，事实上把OPTIONSVerbHandler删除了才会造成OPTIONS失败！  
难道是部分环境下会这样？

总结一下我的解决方案(IIS10 + asp.net web api)：  
1、IIS中增加响应头  
Access-Control-Allow-Origin：\*  
Access-Control-Allow-Headers：Content-Type  
2、web.config中把“<remove name="OPTIONSVerbHandler" />”删掉

仅此而已~

2017年7月 5日 10:50 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-378484) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**~~~~**](http://xn--gzrv62bz6d/)**说：**

post请求变成了options，请问该怎么设置呢？

2017年7月17日 11:47 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-378730) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**刘建 说：**

我按照上面配置的，怎么还报这个错误，求指教  
Refused to set unsafe header "Access-Control-Request-Method"

2017年7月21日 18:03 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-378835) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**前端菜鸟 说：**

复杂请求Content-Type必须为application/json吗还是可以为其他的

2017年7月21日 18:06 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-378836) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**weiwei 说：**

IE10以下怎么办

2017年8月 3日 11:45 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-379205) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**xiongzhe 说：**

恍然大悟，学习了！

2017年8月 4日 09:40 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-379228) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**王叶深**](http://greedying.com/)**说：**

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

这里应该是 服务器否定了预检请求吧。

请教下，如果服务器否定了预检请求，那应该返回什么呢？ 状态码应该是403还是200？是否继续往下执行程序？谢谢

2017年8月 4日 15:29 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-379240) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**xhuanhuan 说：**

想请教阮大大关于cors跨域的安全问题，Access-Control-Allow-Origin限定了域名的情况下，还会有什么安全问题呢？

2017年8月 5日 21:02 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-379284) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**nihao 说：**

ajax不就是用xmlhttprequest实现的吗 为何单拿出来ajax

2017年8月20日 11:19 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-379669) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**伯格 说：**

不对啊！前一个文章里您是这么说的。  
“同源政策规定，AJAX请求只能发给同源的网址，否则就报错。”

按您的说法，跨域了就直接报错了，怎么会发出去呢？

而且我自己实验了一下，我们公司的接口都是可以跨域的，访问方法和平时发送ajax请求没区别。

所以是不是所谓跨域不能请求ajax指的是，后端通过Access-Control-Allow-Origin来限制跨域请求，而不是浏览器限制你发送ajax跨域请求？

2017年9月22日 17:37 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-380754) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**lant**](http://wu/)**说：**

cors简单请求不是说不能超过下面的字段么:  
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

第一个简单请求的例子中就有  
Host  
Connection  
User-Agent  
这些都不再上面的字段中啊, 怎么理解阮大神说的 不能超过如下字段

2017年9月26日 15:08 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-380893) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**lz 说：**

老师啊, 我前端端口是8699，后台端口是8088,后台登录请求里面设置cookie是不是设置不了啊？为什么下次后台读取cookie的时候，读取不到cookie

2017年11月 2日 15:57 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-381921) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**Ryze 说：**

干货充足，没有太多的废话，赞^32!

2017年11月23日 14:59 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-382655) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**残心 说：**

阮老师您好：  
我想咨询您个问题，当我进行跨域的post请求，这个请求为非简单请求的时候，我想要携带cookie，浏览器会做出一次预检请求，但是这次预检请求是并不携带cookie的，这就导致了服务端又新生成了一个cookie（然而这个cookie并用不上），请问一下是不是我哪里操作错误了还是其他什么情况~  
fetch("http://localhost:3000/login", {  
method: "post",   
mode: "cors",  
credentials: "include",   
headers: {'Content-Type': "application/json"}  
});  
同理使用jq的ajax也一样  
$.ajax({  
url: 'http://localhost:3000/login',  
type: 'post',  
xhrFields: {  
withCredentials: true  
},  
headers:{'Content-Type': "application/json"}  
});  
并且jq请求回来的cookies在控制台里面看不到，当再次请求的时候却不知道缘何携带上了上次返回设置的cookie。。。但是fetch的请求可以在控制台看到返回的设置的cookie。。。  
不知道问题出在哪里，烦请老师解答，不胜感激~

2017年12月14日 14:46 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-383227) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**df**](http://dddd/)**说：**

dfsd

2017年12月21日 14:35 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-383421) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**hzj629206 说：**

@伯格：

你用的浏览器已经支持CORS了吧。不支持CORS的浏览器应该是发不出去。

2018年1月 5日 11:27 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-384136) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**Lx13 说：**

帮助挺大的，谢谢阮老师

2018年1月19日 13:30 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-384726) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**许杰 说：**

@阮一峰，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。  
这里面有个错误吧，应该说服务器否定了“预检”请求，怎么变成浏览器了。

2018年2月 6日 11:08 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-385578) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**HGW-XX 说：**

服务器是不会对什么预检查请求做否定的，仍然会正常200返回，只是一旦浏览器发现返回中无跨域许可的相关设置，就可在浏览器的console中报异常了，也就不会再发正式的请求了，所以是浏览器自己把自己给否定了

2018年2月26日 20:33 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-386033) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**HGW-XX 说：**

**引用久保与二弦琴的发言：**

您好，在测试这个问题的时候，发现即便服务器端不设置Access-Control-Allow-Origin头，服务器依然会返回数据，可以用fiddler抓取到，和设置头信息后的区别只是浏览器页面不显示数据而已。能否表明，CORS设置与否只是影响浏览器是否显示服务器返回的数据，而无法阻止服务器要不要返回数据?

我觉得您说的很对，好像就是这么一回事，我之前自己简单测试时也发现这个情况，所以我认为，跨域请求只不过是浏览器自己给自己划得底线和操守…，要真是它觉得有没有access-control-allow-origin也无所谓的话，所不定照样请求过去，也没有什么问题

2018年2月26日 20:37 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-386034) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**HGW-XX 说：**

阮老师，您好，看了您的这篇文章，感觉跨域请求的整个来龙去脉，算是基本明白了，

但是还有一个问题，是不是 简单请求的情况下，就算服务器返回的响应头中，没有access-control-allow-origin的设置，但后端的服务并不会受影响，后端仍然会针对“简单请求”提供服务，只不过是浏览器自己比较高分亮节，不去处理返回的数据，

是这样的吗，那跨域请求岂不是对于后端来说，仍然不是很安全， 除非在 比如Spring的filter中将跨域的请求进行过滤才行？

2018年2月26日 20:42 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-386035) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**刘曦光**](http://g10guang.com/)**说：**

在介绍简单请求和复杂请求的地方：  
Content-Language 是不是打错了？应该是 Content-Length

2018年3月22日 10:05 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-386946) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**lindahuo**](https://github.com/QiudiLin)**说：**

阮老师，关于您文章里面说的：“Access-Control-Allow-Credentials 的值表示是否允许发送 Cookie”，我在 MDN 中文版本里面没看到有这个意思啊，MDN 只是说：“它指定了当浏览器的 credentials 设置为 true 时是否允许浏览器读取 response 的内容，同时简单 GET 请求如果响应中不包含该字段，这个响应将被忽略掉”。

2018年4月 3日 17:10 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-387444) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**Ak\_allen 说：**

Access-Control-Expose-Headers 里面加了 Set-cookie 的白名单，不过客户端 ajax xhr.getResponseHeader\*("") 获取到还是 null 怎么回事

2018年4月 4日 07:57 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-387459) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**chj 说：**

阮老师你好，关于这个跨域我最近有个问题  
我电脑本机上用vue-cli的开发环境，去请求域名服务器的接口，我设置了  
Axios.defaults.withCredentials = true;  
后端是php代码  
if(isset($\_SERVER["HTTP\_ORIGIN"])) {  
header('Access-Control-Allow-Origin:'.$\_SERVER["HTTP\_ORIGIN"]);   
}  
header('Access-Control-Allow-Methods:OPTIONS, GET, POST');  
header('Access-Control-Allow-Headers:x-requested-with');  
header('Access-Control-Max-Age:86400');  
header('Access-Control-Allow-Credentials:true');  
但是我在chrome network 中查看request header 还是没有发现携带cookie  
关于这方面，我看了其他的文章有说设置domain 但是document.domain 又被禁止，document.cookie = "domain=xxx.com"（xxx.com 是我请求的接口的网站域名）依旧不行  
我在这些设置中有些混乱，我始终没有把我本地的cookie 携带过去，后端要接受cookie 在登录状态判断。  
请教阮老师。我是否哪里遗漏了，还是做错了？还需要什么配置吗？谢谢阮老师

2018年4月14日 16:27 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-387985) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**周周 说：**

@甘训奏：

这可能是在服务器没有提供对Options请求的处理，参考：[Spring mvc解决跨域请求：Response to preflight request doesn't pass access control check](https://majing.io/posts/10000010111185)

2018年4月16日 09:52 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-388021) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**潘明杰 说：**

老师  
我发送的请求时origin是null怎么解

2018年4月17日 14:42 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-388047) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**柴全起 说：**

阮老师你好，我想问下，后端设置CORS了，但是还是报错，好像是浏览器拦截了，前端需要设置xhr.withCredentials = true就行了吗？

2018年4月23日 12:02 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-388181) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**迪壳 说：**

阮老师,您好  
关于 Access-Control-Allow-Credentials 的有一部分不太理解  
有提到说 Access-Control-Allow-Credentials为true的时候才发送cookie  
但是这个时候不是已经接收到服务器的回应了吗? 这样的话第一次请求不是无法携带cookie吗?

2018年7月11日 22:59 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-390826) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**ericxu1983**](http://www.xuminggang.com/)**说：**

这篇文章写的不太好，找了一篇英文的tutorial,写的更好一些：https://www.html5rocks.com/en/tutorials/cors/

2018年7月17日 10:33 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-391018) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**besthyhy 说：**

Spring Boot 一个 @CrossOrigin 就可以跨域访问了。 但是本地调试的时候，https://localhost:63343 的页面访问不通 [https://localhost:8488的接口。](https://localhost:8488%E7%9A%84%E6%8E%A5%E5%8F%A3%E3%80%82/)

有办法吗？感激不尽！

2018年7月19日 00:37 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-391103) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**JavanLu 说：**

"If request method is not a case-sensitive match for any method in methods and is not a simple method, apply the cache and network error steps."   
  
----参见 <https://www.w3.org/TR/cors/#preflight-request>

由于 GET | POST | HEAD 是简单方法，在通过预检请求后，即使 Access-Control-Allow-Methods 中没有返回 GET | POST | HEAD , 这三种类型的非简单请求仍然会成功。

2018年7月26日 16:32 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-391436) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

**小心鲨鱼 说：**

阮老师，我们公司后端验证登录系统，验证所有的请求的cookie，但是options请求没有携带cookie所以直接被重定向到了登录界面，所以option请求是否可以携带cookie吗？

2018年8月 1日 10:15 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-391679) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**Michael**](http://limingzhi198@163.com/)**说：**

写的结构明确易读，思路很清晰，感觉以后搜索的时候要先看阮老师有没有相关的文章了。  
建议把下方 的评论折叠一下，看到进度条好长，差点被吓跑了。哈哈...

2018年8月27日 10:09 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-392585) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)

[**刘曦光**](https://g10guang.com/)**说：**

跨越方面也可以看看这文章:https://segmentfault.com/a/1190000011145364

2018年9月18日 14:26 | [#](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-393224) | [引用](http://www.ruanyifeng.com/blog/2016/04/cors.html#comment-text)