Developer RP 799 发布于 全栈开发之路
2017-12-23 发布

Nginx配置跨域请求 Access-Control-Allow-Origin *

当出现403跨域错误的时候 `No 'Access-Control-Allow-Origin' header is present on the requested resource`，需要给Nginx服务器配置响应的header参数：

一、 解决方案

只需要在Nginx的配置文件中配置以下参数：

```
location / {
    add_header Access-Control-Allow-Origin *;
    add_header Access-Control-Allow-Methods 'GET, POST, OPTIONS';
    add_header Access-Control-Allow-Headers 'DNT,X-Mx-ReqToken,Keep-Alive,User-Agent,X-Re

    if ($request_method = 'OPTIONS') {
        return 204;
    }
}
```

> 上面配置代码即可解决问题了，不想深入研究的，看到这里就可以啦=-=

二、 解释

1. Access-Control-Allow-Origin

服务器默认是不被允许跨域的。给Nginx服务器配置`Access-Control-Allow-Origin *`后，表示服务器可以接受所有的请求源（Origin），即接受所有跨域的请求。

2. Access-Control-Allow-Headers 是为了防止出现以下错误：

`Request header field Content-Type is not allowed by Access-Control-Allow-Headers in preflight response.`

这个错误表示当前请求Content-Type的值不被支持。其实是我们发起了"application/json"的类型请求导致的。这里涉及到一个概念：`预检请求（preflight request）`,请看下面"预检请求"的介绍。

3. **Access-Control-Allow-Methods** 是为了防止出现以下错误：

```
Content-Type is not allowed by Access-Control-Allow-Headers in preflight response.
```

4.给`OPTIONS` 添加 `204`的返回，是为了处理在发送POST请求时Nginx依然拒绝访问的错误

发送"预检请求"时，需要用到方法 `OPTIONS` ,所以服务器需要允许该方法。

三、 预检请求（preflight request）

其实上面的配置涉及到了一个W3C标准：`CROS`,全称是跨域资源共享 (Cross-origin resource sharing)，它的提出就是为了解决跨域请求的。

> 跨域资源共享(CORS)标准新增了一组 HTTP 首部字段，允许服务器声明哪些源站有权限访问哪些资源。另外，规范要求，`对那些可能对服务器数据产生副作用的HTTP 请求方法（特别是 GET 以外的 HTTP 请求，或者搭配某些 MIME 类型的 POST 请求）`，浏览器必须首先使用 OPTIONS 方法发起一个预检请求（preflight request），从而获知服务端是否允许该跨域请求。服务器确认允许之后，才发起实际的 HTTP 请求。在预检请求的返回中，服务器端也可以通知客户端，是否需要携带身份凭证（包括 Cookies 和 HTTP 认证相关数据）。

其实`Content-Type字段的类型为application/json`的请求就是上面所说的`搭配某些 MIME 类型的 POST 请求`,CORS规定，Content-Type不属于以下MIME类型的，都属于预检请求：

```
application/x-www-form-urlencoded
multipart/form-data
text/plain
```

所以 application/json的请求 会在正式通信之前，增加一次"预检"请求，这次"预检"请求会带上头部信息`Access-Control-Request-Headers: Content-Type`：

```
OPTIONS /api/test HTTP/1.1
Origin: http://foo.example
Access-Control-Request-Method: POST
Access-Control-Request-Headers: Content-Type
... 省略了一些
```

服务器回应时，返回的头部信息如果不包含`Access-Control-Allow-Headers: Content-Type`则表示不接受非默认的的Content-Type。即出现以下错误：

```
Request header field Content-Type is not allowed by Access-Control-Allow-Headers in preflight response.
```

参考文章：

阮一峰【跨域资源共享 CORS 详解】
MDN web docs【HTTP访问控制（CORS）】

你可能感兴趣的

- **前端跨域策略实践** holyZhengs 跨域 cors jsonp 同源策略 javascript
- **CORS方式实现ajax跨域 --- nginx配置** April linux nginx
- **fetch 跨域请求** THZXQ fetch cors
- **Nginx实践篇（3）- 跨域访问** 白菜1031 运维 webserver nginx linux php
- **CORS跨域资源共享** chenhao_ch cors
- **探究CORS（跨域资源共享）** parvin http cors jsonp
- **Django解决ajax跨域访问问题** rayzz 跨域
- **CORS跨域模型浅析及常见理解误区分析** 木子墨 cors

14 条评论

默认排序 时间排序

Aceslup · 2018年09月06日

倒数第三行是不是写错了。服务器响应的，应该是Access-Control-Allow-Headers吧?

赞 回复

> 确实写错了，改过来了，感谢指正！
> — **Developer** 作者 · 2018年09月07日