

Window下nginx调试命令:

```
# 关闭所有nginx进程
taskkill /IM nginx.exe /F
# 检查 nginx.conf 配置文件
nginx -t
# 启动nginx
start nginx.exe
```

我们在项目中，可能会遇到跨域问题，然后有些问题nginx可以解决，有些不可以解决。

在不可以解决的情况下 只能通过 前端或者后端添加代码来解决。（但是如果前后端配置跨域则nginx不能配置。否则会报错重复问题）

原因：因为Nginx只能对静态类似HTML这种做跨域才会生效。（Nginx里nginx.conf设置的跨域是对整个生效）

所以如果Nginx设置跨域而无法生效就只能从后端中加代码设置跨域来解决。（但是后端如果加了我们则不能直接再在nginx里设置跨域因为会报错重复，但是如果我们不加跨域则会引起别的接口的调用问题）

nginx.conf配置如下:

```
#user nobody;
worker_processes 1;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;

#pid logs/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
```

```

sendfile      on;

keepalive_timeout 65;

gzip on;
server {
    listen      80;
    server_name 服务名;

    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Server $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    location / {
        proxy_pass http://127.0.0.1:要代理的后端端口;
        proxy_connect_timeout 600;
        proxy_read_timeout 600;
    }
}
server {
    listen      80;
    server_name 服务名;

    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Server $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    # 配置后端服务地址
    location /api/upload { # 指定代理的后端api路径
        proxy_pass http://127.0.0.1:要代理的后端端口;
        proxy_connect_timeout 600;
        proxy_read_timeout 600;

        rewrite "^/api/(.*)$" /$1 break;
    }

    location / { # 前面的路径都没匹配到的api访问这里
        proxy_pass http://127.0.0.1:要代理的后端端口;
        proxy_connect_timeout 600;
        proxy_read_timeout 600;
    }
}
}

```