Nginx本身不支持PHP/Python/Perl等语言，但是它可以通过FastCGI来将请求扔给某些语言或框架处理。例如将.php为后缀的请求通过FashCGI交给PHP的一个FastCGI管理器PHP-FPM来处理。

关于proxy\_pass和fastcgi\_pass的区别也就在于一个Nginx本身是否支持。

https://img-blog.csdnimg.cn/20190226094153161.png

**1  概述**

nginx代理通过ngx\_http\_fastcgi\_module这个模块，将收到php程序的请求后就转发到后台FastCGI服务器处理，这里nginx可以把php-fpm服务运行在同一机器上，也可以将nginx和php-fpm分离在两台机器上。但是，nginx不支持php模块方式,只能是php-fpm模式。

本文将介绍ngx\_http\_fastcgi\_module模块的相关命令和配置。

**2  fastcgi配置**

**.1、fastcgi\_pass**

fastcgi\_pass  address;

address为后端的fastcgi  server的地址

可用位置：location,if in location

**.2、fastcgi\_index**

fastcgi\_index  name;

fastcgi默认的主页资源

示例：fastcgi\_index  index.php;

**.3、fastcgi\_param**

fastcgi\_param  parameter value [if\_not\_empty];

设置传递给FastCGI服务器的参数值，可以是文本，变量或组合

# proxy\_pass和fastcgi\_pass区别，反向代理，负载均衡、nginx+fastcgi+php的并发阻塞问题

**目录**

[1、准备两个tomcat和一个nginx](https://blog.csdn.net/xixiyuguang/article/details/88891698#1%E3%80%81%E5%87%86%E5%A4%87%E4%B8%A4%E4%B8%AAtomcat%E5%92%8C%E4%B8%80%E4%B8%AAnginx)

[1.1、tomcat: http://127.0.0.1:8080](https://blog.csdn.net/xixiyuguang/article/details/88891698#1.1%E3%80%81tomcat%3A%20http%3A%2F%2F127.0.0.1%3A8080)

[1.2、tomcat:http://127.0.0.1:8081](https://blog.csdn.net/xixiyuguang/article/details/88891698#1.2%E3%80%81tomcat%3Ahttp%3A%2F%2F127.0.0.1%3A8081)

[1.3、nginx:http://127.0.0.1](https://blog.csdn.net/xixiyuguang/article/details/88891698#1.3%E3%80%81nginx%3Ahttp%3A%2F%2F127.0.0.1)

[2、反向代理proxy\_pass](https://blog.csdn.net/xixiyuguang/article/details/88891698#1%E3%80%81%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86http%20proxy%3A)

[2.1、反向代理](https://blog.csdn.net/xixiyuguang/article/details/88891698#2.1%E3%80%81%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86)

[2.2、反向代理-使用upstream](https://blog.csdn.net/xixiyuguang/article/details/88891698#2.2%E3%80%81%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86-%E4%BD%BF%E7%94%A8upstream)

[2.3、反向代理-负载均衡-轮循模式](https://blog.csdn.net/xixiyuguang/article/details/88891698#2.3%E3%80%81%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86-%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1-%E8%BD%AE%E5%BE%AA%E6%A8%A1%E5%BC%8F)

[2.4、反向代理-负载均衡-权重模式](https://blog.csdn.net/xixiyuguang/article/details/88891698#2.4%E3%80%81%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86-%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1-%E6%9D%83%E9%87%8D%E6%A8%A1%E5%BC%8F)

[2.4、反向代理-负载均衡-ip\_hash方式](https://blog.csdn.net/xixiyuguang/article/details/88891698#2.4%E3%80%81%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86-%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1-ip_hash%E6%96%B9%E5%BC%8F)

[3、网关代理fastcgi\_pass（php+nginx配合）](https://blog.csdn.net/xixiyuguang/article/details/88891698#2%E3%80%81fastcgi_pass%C2%A0%C2%A0%20%3A)

[3.1、php安装（本人安装的是php-7.3.1）](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.1%E3%80%81php%E5%AE%89%E8%A3%85%EF%BC%88%E6%9C%AC%E4%BA%BA%E5%AE%89%E8%A3%85%E7%9A%84%E6%98%AFphp-7.3.1%EF%BC%89)

[3.1.1、php下载](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.1.1%E3%80%81php%E4%B8%8B%E8%BD%BD)

[3.1.2、php配置](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.1.2%E3%80%81php%E9%85%8D%E7%BD%AE)

[3.2、开启cgi端口](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.2%E3%80%81%E5%BC%80%E5%90%AFcgi%E7%AB%AF%E5%8F%A3)

[3.3、fastcgi\_passnginx配置](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.3%E3%80%81fastcgi_passnginx%E9%85%8D%E7%BD%AE)

[3.3.1、新建文件夹：E:\php\www  。然后创建两个文件](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.3.1%E3%80%81%E6%96%B0%E5%BB%BA%E6%96%87%E4%BB%B6%E5%A4%B9%EF%BC%9AE%3A%5Cphp%5Cwww%C2%A0%20%E3%80%82%E7%84%B6%E5%90%8E%E5%88%9B%E5%BB%BA%E4%B8%A4%E4%B8%AA%E6%96%87%E4%BB%B6)

[3.3.2、创建文件index.php](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.3.2%E3%80%81%E5%88%9B%E5%BB%BA%E6%96%87%E4%BB%B6index.php)

[3.3.3、创建文件index.html](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.3.3%E3%80%81%E5%88%9B%E5%BB%BA%E6%96%87%E4%BB%B6index.html)

[3.3.4、修改nginx配置](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.3.4%E3%80%81%E4%BF%AE%E6%94%B9nginx%E9%85%8D%E7%BD%AE)

[3.4、调试页面截图](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.4%E3%80%81%E8%B0%83%E8%AF%95%E9%A1%B5%E9%9D%A2%E6%88%AA%E5%9B%BE)

[3.5、解决 nginx+fastcgi+php的并发阻塞问题](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.5%E3%80%81%E8%A7%A3%E5%86%B3%20nginx%2Bfastcgi%2Bphp%E7%9A%84%E5%B9%B6%E5%8F%91%E9%98%BB%E5%A1%9E%E9%97%AE%E9%A2%98)

[3.5.1、启动多个cgi端口（9000，9001端口）](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.5.1%E3%80%81%E5%90%AF%E5%8A%A8%E5%A4%9A%E4%B8%AAcgi%E7%AB%AF%E5%8F%A3%EF%BC%889000%EF%BC%8C9001%E7%AB%AF%E5%8F%A3%EF%BC%89)

[3.5.2、nginx配置](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.5.2%E3%80%81nginx%E9%85%8D%E7%BD%AE)

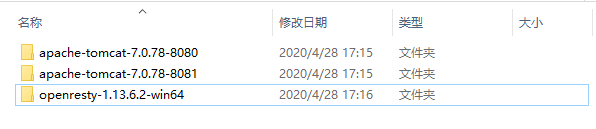
[3.5.3、测试](https://blog.csdn.net/xixiyuguang/article/details/88891698#3.5.3%E3%80%81%E6%B5%8B%E8%AF%95)

[4、不同项目地址转发](https://blog.csdn.net/xixiyuguang/article/details/88891698#4%E3%80%81%E4%B8%8D%E5%90%8C%E9%A1%B9%E7%9B%AE%E5%9C%B0%E5%9D%80%E8%BD%AC%E5%8F%91)

[4.1、在location层转发](https://blog.csdn.net/xixiyuguang/article/details/88891698#4.1%E3%80%81%E5%9C%A8location%E5%B1%82%E8%BD%AC%E5%8F%91)

[4.2、在location里面判断转发](https://blog.csdn.net/xixiyuguang/article/details/88891698#4.2%E3%80%81%E5%9C%A8location%E9%87%8C%E9%9D%A2%E8%BD%AC%E5%8F%91)

**1、准备两个tomcat和一个nginx**



**1.1、tomcat:**[http://127.0.0.1:8080](http://127.0.0.1/)



**1.2、tomcat:**[http://127.0.0.1:8081](http://127.0.0.1/)



**1.3、nginx:**[http://127.0.0.1](http://127.0.0.1/)

**2、反向代理proxy\_pass**

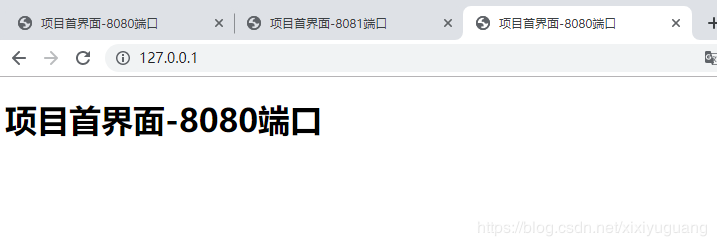
官方文档;<http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_pass>

**2.1、反向代理**

介绍：127.0.0.1的80端口转发到127.0.0.1:8080. 。在这里127.0.0.1就类似于服务器的ip地址同理。

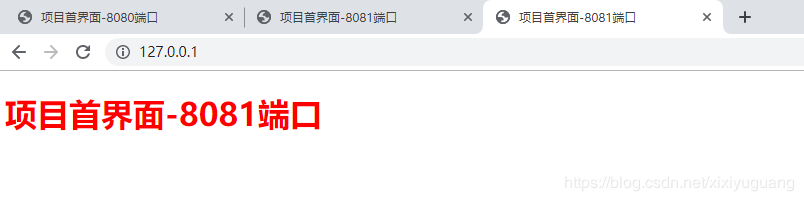
1. server {
2. listen 80;
3. server\_name localhost www.xxx.com; #域名
4. location / {
5. proxy\_pass http:*//127.0.0.1:8080;*
6. client\_max\_body\_size 1024m; #请求体大小设置
7. }
8. }

效果展示



**2.2、反向代理-使用upstream**

1. upstream tomcat{
2. server 127.0.0.1:8081;
3. }
5. server {
6. listen 80;
7. server\_name localhost www.xxx.com; #域名
8. location / {
9. proxy\_pass http:*//tomcat;*
10. client\_max\_body\_size 1024m; #请求体大小设置
11. }
12. }



**2.3、反向代理-负载均衡-轮循模式**

轮询方式是默认方式，每个请求按照时间顺序逐一分配到不同的后端服务器进行处理，如有服务器宕机会被自动剔除。

1. # 负载均衡-轮循模式
2. upstream tomcat{
3. server 127.0.0.1:8080;
4. server 127.0.0.1:8081;
5. }
7. server {
8. listen 80;
9. server\_name localhost www.xxx.com; #域名
10. location / {
11. proxy\_pass http:*//tomcat;*
12. client\_max\_body\_size 1024m; #请求体大小设置
13. }
14. }

访问的时候刷新，会发现每次都不一样。为轮循

**2.4、反向代理-负载均衡-权重模式**

权重方式是利用weight指定的权重比率，与访问率成正比。用于后端服务器性能不均的情况。

#  weigth参数表示权值，权值越高则被分配到的概率越大。测试的时候发现weight=3，刷新的时候8080端口就刷出3次

#  max\_fails允许请求失败的次数默认为l

#  fail\_timeoutq请求失败后暂停服务的时间

#  backup预留的备份机器

#  down表示当前的server暂时不参与负载均衡。

1. # weigth参数表示权值，权值越高则被分配到的概率越大。测试的时候发现weight=3，刷新的时候8080端口就刷出3次
2. # max\_fails允许请求失败的次数默认为l
3. # fail\_timeoutq请求失败后暂停服务的时间
4. # backup预留的备份机器
5. # down表示当前的server暂时不参与负载均衡。
6. upstream tomcat{
7. #weigrt权重值(越大访问率大),在fail\_timeout时间内检查后端服务器max\_fails次，失败则被剔除;
8. server 127.0.0.1:8080 weight=3 fail\_timeout=30s max\_fails=2;
9. server 127.0.0.1:8081 weight=1 fail\_timeout=30s max\_fails=2;
10. }
12. server {
13. listen 80;
14. server\_name localhost www.xxx.com; #域名
15. location / {
16. proxy\_pass http:*//tomcat;*
17. client\_max\_body\_size 1024m; #请求体大小设置
18. }
19. }

**2.4、反向代理-负载均衡-ip\_hash方式**

ip\_hash方式是按每个请求访问IP的hash结果分配，可以使每个访客固定访问一个后端服务器，可以解决Session共享的问题。

一般情况下不太建议使用这种方式

1. *# ip\_hash;*
2. upstream tomcat{
3. ip\_hash;
4. server 127.0.0.1:8080 fail\_timeout=5s ;
5. server 127.0.0.1:8081 fail\_timeout=5s ;
6. }
8. server {
9. listen 80;
10. server\_name localhost www.xxx.com; #域名
11. location / {
12. proxy\_pass http://tomcat;
13. client\_max\_body\_size 1024m; #请求体大小设置
14. }
15. }

**3、网关代理fastcgi\_pass（php+nginx配合）**

[官方文档：    http://nginx.org/en/docs/http/ngx\_http\_fastcgi\_module.html#fastcgi\_pass](http://nginx.org/en/docs/http/ngx_http_fastcgi_module.html#fastcgi_pass)

    fastcgi介绍：CGI全称是“[公共网关接口](http://baike.baidu.com/view/649193.htm)”(Common Gateway Interface)，HTTP服务器与你的或其它机器上的程序进行“交谈”的一种工具，其程序一般运行在[网络服务器](http://baike.baidu.com/view/813.htm)上。 CGI可以用任何一种语言编写，只要这种语言具有标准输入、输出和[环境变量](http://baike.baidu.com/view/95930.htm)。如[php](http://baike.baidu.com/view/99.htm),[perl](http://baike.baidu.com/view/46614.htm),tcl等。

这种模式一般和php配合一起使用所以我们首先要安装php

**3.1、php安装（本人安装的是php-7.3.1）**

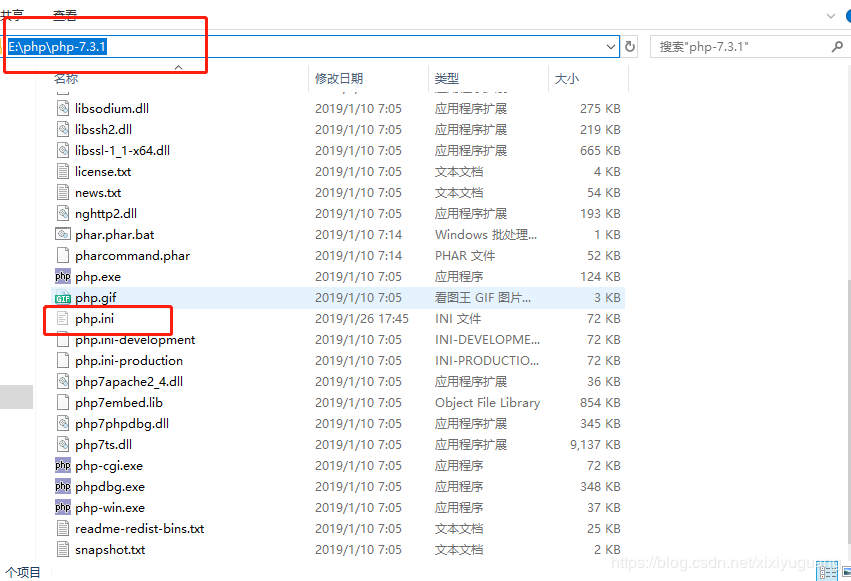
**3.1.1、php下载**

进入PHP下载地址<http://windows.php.net/download>下载最新线程安全版PHP zip压缩包，解压缩后放在想要安装的路径下。注意：下载的PHP VC版本不能比前面安装的vc redist版高。

**3.1.2、php配置**

进入PHP安装目录，复制一份php.ini-development改名为php.ini放到安装路径下，打开找到"extension\_dir"，去掉注释符，将值改为"PHP安装路径\ext"。我当前的是：extension\_dir = "E:\php\php-7.3.1\ext"

如图：



**3.2、开启cgi端口**

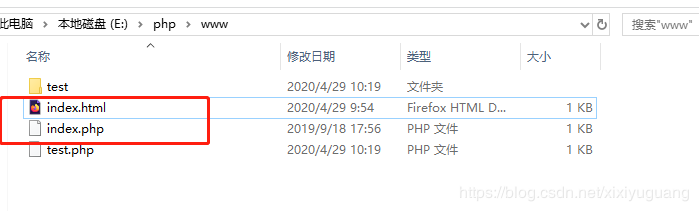
1. *// 命令，在控制台执行*
2. E:\php\php-7.3.1\php-cgi.exe -b 127.0.0.1:9000 -c E:\php\php-7.3.1\php.ini

如图：



**3.3、fastcgi\_passnginx配置**

**3.3.1、新建文件夹：E:\php\www  。然后创建两个文件**



**3.3.2、创建文件index.php**

1. <?php
2. echo '<strong>Hello, this is index.php!</strong>';
3. ?>

**3.3.3、创建文件index.html**

this is index html

**3.3.4、修改nginx配置**

1. server {
2. listen 7000;
3. server\_name localhost;
4. location / {
5. *## 解决跨域的代码*
6. add\_header Access-Control-Allow-Origin \*;
7. add\_header Access-Control-Allow-Headers X-Requested-With;
8. add\_header Access-Control-Allow-Methods GET,POST,PUT,DELETE,OPTIONS;
9. *#创建的包含 index.html和 index.php的文件夹*
10. root E:/php/www;
11. index index.html index.htm index.php;
12. try\_files $uri $uri/ /index.php?$query\_string;
13. }
15. *# 判断后缀为php的地址*
16. location ~ \.php$ {
17. *#创建的包含 index.html和 index.php的文件夹*
18. root E:/php/www;
19. fastcgi\_pass 127.0.0.1:9000; *#端口是上面开通的cgi端口*
20. fastcgi\_index index.php;
21. fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;
22. include fastcgi\_params;
23. }
24. error\_page 500 502 503 504 /50x.html;
25. }

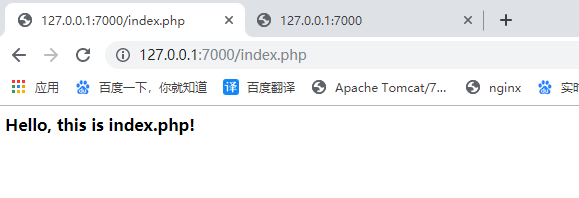
重启nginx：nginx.exe -s reload

**3.4、调试页面截图**

访问：<http://127.0.0.1:7000/>



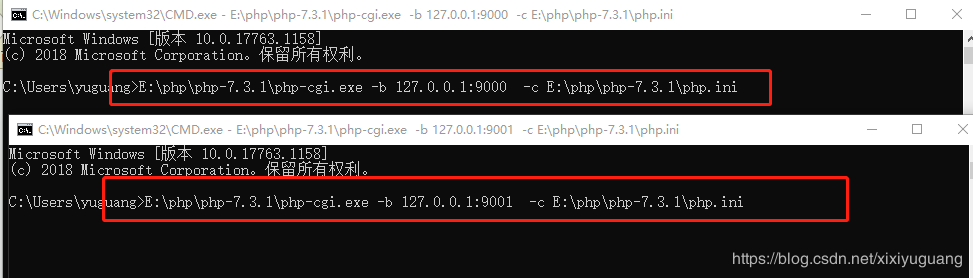
访问：<http://127.0.0.1:7000/index.php>



**3.5、解决 nginx+fastcgi+php的并发阻塞问题**

**3.5.1、启动多个cgi端口（9000，9001端口）**

命令：E:\php\php-7.3.1\php-cgi.exe -b 127.0.0.1:9000  -c E:\php\php-7.3.1\php.ini



**3.5.2、nginx配置**

然后重启nginx：E:\php\nginx-1.18.0>  **nginx.exe -s reload**

1. upstream fastcgi\_proxy{
2. server 127.0.0.1:9000;
3. server 127.0.0.1:9001;
4. }
6. server {
7. listen 7000;
8. server\_name localhost;
9. location / {
10. add\_header Access-Control-Allow-Origin \*;
11. add\_header Access-Control-Allow-Headers X-Requested-With;
12. add\_header Access-Control-Allow-Methods GET,POST,PUT,DELETE,OPTIONS;
13. root E:/php/www;
14. index index.html index.htm index.php;
15. try\_files $uri $uri/ /index.php?$query\_string;
16. }
18. location ~ \.php$ {
19. root E:/php/www;
20. fastcgi\_pass fastcgi\_proxy;
21. fastcgi\_index index.php;
22. fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;
23. include fastcgi\_params;
24. client\_max\_body\_size 10m;
25. }
26. error\_page 500 502 503 504 /50x.html;
27. }

**3.5.3、测试**

经过测试发现，9000端口和9001端口只要有一个还在启动，[http://127.0.0.1:7000/index.php 这个地址就可以访问](http://127.0.0.1:7000/index.php)

如由于需要相关软件和资料，欢迎留言

**4、不同项目地址转发**

**4.1、在location层转发**

1. server {
2. listen 8088;
3. server\_name localhost;
4. location ^~ /crm {
5. proxy\_pass http://127.0.0.1:8080/crm;
6. }
7. location ^~ /resourcesSystem {
8. proxy\_pass http://127.0.0.1:8080/resourcesSystem;
9. }
10. location ^~ /sw\_agency {
11. proxy\_pass http://127.0.0.1:8080/sw\_agency;
12. }
14. }

## 4.2、在location里面判断转发

1. server {
2. listen 8088;
3. server\_name localhost;
4. location / {
5. proxy\_http\_version 1.1;
6. proxy\_set\_header Upgrade $http\_upgrade;
7. proxy\_set\_header Connection "Upgrade";
9. root html;
10. index index.html index.htm;
12. if ($request\_uri ~\* "^\/springboot"){
13. *# add\_header Access-Control-Allow-Origin \*;*
14. *# add\_header Access-Control-Allow-Headers X-Requested-With,content-type,cert,t;*
15. *# add\_header Access-Control-Allow-Methods GET,POST,OPTIONS;*
16. proxy\_pass http://127.0.0.1:8080;break;
17. }
18. }
19. }