# 基本命令

windows: 进入到nginx 主文件夹路径下面

nginx -s stop 快速停止nginx

nginx -s reload 热重启

start nginx(启动windows)

nginx -c /etc/nginx/nginx.conf 按照特定配置文件 启动Linux

# 功能

## 反向代理多端口

可以配置多个监听端口设置不同的规则：如下就可以访问

<http://localhost> 🡪 index.html

<http://localhost:8088> 🡪 <http://localhost:8082/recordZs/>

通过配置proxy\_pass 设置服务器跳转

server {

listen 80;

server\_name localhost;

location / {

root html;

index index.html index.htm;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

server {

listen 8088;

server\_name recordZs;

location / {

proxy\_pass http://localhost:8082/recordZs/;

}

}

## 反向代理location规则

server {

listen 8088;

server\_name recordZs;

location / {

proxy\_pass http://localhost:8082/recordZs/;

}

}

匹配成功之后，所有的 8088/的请求 都会转到 以<http://localhost:8082/recordZs/>开头的请求中

例如 localhost:8088/index.jsp 🡪 <http://localhost:8082/recordZs/index.jsp>

location匹配命令

~ #波浪线表示执行一个正则匹配，区分大小写（ \----$ 局部匹配 ^----$ 打头和末位匹配 ）

~\* #表示执行一个正则匹配，不区分大小写

^~ #^~表示普通字符匹配，如果该选项匹配，只匹配该选项，不匹配别的选项，一般用来匹配目录

= #进行普通字符精确匹配

@ #"@" 定义一个命名的 location，使用在内部定向时，例如 error\_page, try\_files

$ 引用变量（$server\_name表示引用一串地址）

server {

listen 80;

server\_name edu.gehealthcare.cn;

location ~ ^(/index)$ {

return 301 http://$server\_name/edu\_index;

}

}

location = / {

# 只匹配"/".

[ configuration A ]

}

location / {

# 匹配任何请求，因为所有请求都是以"/"开始

# 但是更长字符匹配或者正则表达式匹配会优先匹配

[ configuration B ]

}

location ^~ /images/ {

# 匹配任何以 /images/ 开始的请求，并停止匹配 其它location

[ configuration C ]

}

location ~\* .(gif|jpg|jpeg)$ {

# 匹配以 gif, jpg, or jpeg结尾的请求.

# 但是所有 /images/ 目录的请求将由 [Configuration C]处理.

[ configuration D ]

}

## 动静分离匹配

注意事项：

1. windows的话 路径一定是正斜杠” / “， 而不能是 反斜杠 啊
2. 静态资源的话，用 root 开头
3. 静态资源的开头要按照规则 IP:8088/RESIRYSE

server {

listen 8088;

server\_name recordZs;

location ~ \.(gif|jpg|jpeg|png|bmp|swf)$ {

root C:/Users/502764158/Desktop/test;

}

location ~ \.(jsp|do)$ {

proxy\_pass http://localhost:8082;

}

}

静态资源代码 ip为 反向代理的监听端口 8088，而不是tomcat的端口

<img alt="testNginx" src="http://localhost:8088/images/123.png">

~ 表示匹配正则表达式 \开始匹配 $表示结束匹配

注意正则表达式匹配的话 proxy\_pass中不能包含 URL 元素，也就是 只有 http:\*\*\*:8080 后面不能有元素

# 所有静态请求都由nginx处理，存放目录为html

location ~ \.(gif|jpg|jpeg|png|bmp|swf)$ {

root html;

}

# 所有动态请求都转发给tomcat处理

location ~ \.(jsp|do)$ {

proxy\_pass http://local\_tomcat;

}

# 匹配所有以recordZs 打头的请求

location ~ ^recordZs\*$ {

proxy\_pass http://localhost:8082;

}