



nginx

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # HTTP to HTTPS
    rewrite ^(.*) https://$host$1 permanent;
}

server {
    # SSL
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    # SSL certificates
    ssl_certificate /etc/nginx/ssl/nginx.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx.key;

    # ... }
}
```

proxy

```
location / {
    proxy_pass_header Server;
    proxy_redirect off;

    # Proxy headers
    proxy_set_header Host $Host;
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-scheme $scheme;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Host $server_name;
    proxy_pass https://survey.dreamteenet.com;

    # 404 handling
    try_files $uri $uri/ =404;
}
```

SSL

```

ssl_protocols TLSv1.2 TLSv1.3;
ssl_prefer_server_ciphers on;
ssl_ciphers EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH:RSA+AES128:RSA+AES256;
ssl_ecdh_curve secp384r1;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
resolver 168.95.1.1 168.95.192.1 valid=300s;
resolver_timeout 5s;
add_header X-Frame-Options SAMEORIGIN;
add_header X-Content-Type-Options nosniff;
ssl_dhparam /etc/ssl/certs/dhparam2.pem;

```

proxy_redirect

```

location /one/ {
    proxy_pass      http://upstream:port/kevin/;
    proxy_redirect  default;
}

location /one/ {
    proxy_pass      http://upstream:port/kevin/;
    proxy_redirect  http://upstream:port/kevin/ /one/;
}

```



```

http{} #
server{} #
location{} #

```

nginx

```

server {
    ...
    client_max_body_size 10M;
    ...
}

```

限制2MBphp-fpm

```
client_max_body_size 0;
client_body_timeout 300s
```

变量

```
$host      主机名 Host 主机名
$request_method  请求方法 GET/POST
$remote_addr  客户端 IP 地址
$args        请求参数
$content_length  请求内容长度
$http_user_agent  客户端 agent
$http_cookie  客户端 cookie
$remote_addr  客户端 IP 地址
$remote_port  客户端端口
$server_protocol  请求协议 HTTP/1.0/HTTP/1.1
$server_addr  服务器地址
$server_name  服务器名称
$server_port  服务器端口
```

location

```
location [=|~|~*|^~] /uri/ { ... }
location = /uri      = 完全匹配
location ^~ /uri     ^~ 匹配URL
location ~ pattern   ~ 正则匹配
location ~* pattern  ~* 正则匹配
location /uri        普通匹配
location /           location匹配失败switch默认
```

Nginx url 匹配 /static/20%/aa ^~ /static/ /aa

location 配置:

```
location =
location ^~
location
location
location /
location
```

location /dir01 location /dir01/dir02 http://localhost/dir01/

location = / {

echo "A";

}

location = /login {

echo "B";

}

location ^~ /static/ {

echo "C";

}

location ^~ /static/files {

echo "X";

}

location ~ \.(gif|jpg|png|js|css)\$ {

echo "D";

}

location ~* \.png\$ {

echo "E";

}

location /img {

echo "Y";

}

location / {

echo "F";

}

location = / {

echo "G";

}

location = / {

echo "H";

}

location = / {

echo "I";

}

location = / {

echo "J";

}

location = / {

echo "K";

}

location = / {

echo "L";

}

location = / {

echo "M";

}

location = / {

echo "N";

}

location = / {

echo "O";

}

location = / {

echo "P";

}

location = / {

echo "Q";

}

location = / {

echo "R";

}

location = / {

echo "S";

}

location = / {

echo "T";

}

location = / {

echo "U";

}

location = / {

echo "V";

}

location = / {

echo "W";

}

location = / {

echo "X";

}

location = / {

echo "Y";

}

location = / {

echo "Z";

}

location = / {

echo "AA";

}

location = / {

echo "AB";

}

location = / {

echo "AC";

}

location = / {

echo "AD";

}

location = / {

echo "AE";

}

location = / {

echo "AF";

}

location = / {

echo "AG";

}

location = / {

echo "AH";

}

location = / {

echo "AI";

}

location = / {

echo "AJ";

}

location = / {

echo "AK";

}

location = / {

echo "AL";

}

location = / {

echo "AM";

}

location = / {

echo "AN";

}

location = / {

echo "AO";

}

location = / {

echo "AP";

}

location = / {

echo "AQ";

}

location = / {

echo "AR";

}

location = / {

echo "AS";

}

location = / {

echo "AT";

}

location = / {

echo "AU";

}

location = / {

echo "AV";

}

location = / {

echo "AW";

}

location = / {

echo "AX";

}

location = / {

echo "AY";

}

location = / {

echo "AZ";

}

location = / {

echo "BA";

}

location = / {

echo "BB";

}

location = / {

echo "BC";

}

location = / {

echo "BD";

}

location = / {

echo "BE";

}

location = / {

echo "BF";

}

location = / {

echo "BG";

}

location = / {

echo "BH";

}

location = / {

echo "BI";

}

location = / {

echo "BJ";

}

location = / {

echo "BK";

}

location = / {

echo "BL";

}

location = / {

echo "BM";

}

location = / {

echo "BN";

}

location = / {

echo "BO";

}

location = / {

echo "BP";

}

location = / {

echo "BQ";

}

location = / {

echo "BR";

}

location = / {

echo "BS";

}

location = / {

echo "BT";

}

location = / {

echo "BU";

}

location = / {

echo "BV";

}

location = / {

echo "BW";

}

location = / {

echo "BX";

}

location = / {

echo "BY";

}

location = / {

echo "BZ";

}

location = / {

echo "CA";

}

location = / {

echo "CB";

}

location = / {

echo "CC";

}

location = / {

echo "CD";

}

location = / {

echo "CE";

}

location = / {

echo "CF";

}

location = / {

echo "CG";

}

location = / {

echo "CH";

}

location = / {

echo "CI";

}

location = / {

echo "CJ";

}

location = / {

echo "CK";

}

location = / {

echo "CL";

}

location = / {

echo "CM";

}

location = / {

echo "CN";

}

location = / {

echo "CO";

}

location = / {

echo "CP";

}

location = / {

echo "CQ";

}

location = / {

echo "CR";

}

location = / {

echo "CS";

}

location = / {

echo "CT";

}

location = / {

echo "CU";

}

location = / {

echo "CV";

}

location = / {

echo "CW";

}

location = / {

echo "CX";

}

location = / {

echo "CY";

}

location = / {

echo "CZ";

}

location = / {

echo "DA";

}

location = / {

echo "DB";

}

location = / {

echo "DC";

}

location = / {

echo "DD";

}

location = / {

echo "DE";

}

location = / {

echo "DF";

}

location = / {

echo "DG";

}

location = / {

echo "DH";

}

location = / {

echo "DI";

}

\$request_uri HTTP
\$status
\$body_bytes_sent
\$bytes_sent
\$connection
\$connection_requests
\$msec
\$pipe HTTP (pipelined) pipe "p"
\$http_referer
\$http_user_agent
\$request_length
\$request_time
\$time_iso8601 ISO8601
\$time_local

nginx

Upstream server Upstream

```
upstream balanceServer {  
    server 10.1.22.33:12345;  
    server 10.1.22.34:12345;  
    server 10.1.22.35:12345;  
}  
  
server {  
    server_name fe.server.com;  
    listen 80;  
    location /api {  
        proxy_pass http://balanceServer;  
    }  
}
```

add_header

```
# HTTP response headers borrowed from Nextcloud `htaccess`  
add_header Referrer-Policy "no-referrer" always;  
add_header X-Content-Type-Options "nosniff" always;  
add_header X-Download-Options "noopen" always;  
add_header X-Frame-Options "SAMEORIGIN" always;  
add_header X-Permitted-Cross-Domain-Policies "none" always;  
add_header X-Robots-Tag "noindex, nofollow"
```

```
always;  
add_header X-XSS-Protection "1; mode=block" always;
```

x-robot-tag

Noindex:

Index:

Follow:

Nofollow:

Noimageindex:

None: noindex nofollow

Noarchive: SERP()

Nocache: noarchive Internet Explorer Firefox

Nosnippet: SERP() Meta

Noodyp/noydir [OBSOLETE]: DMOZ SERP() DMOZ 2017

Unavailable_after:

noindex

X-Robots-Tag: noindex

nofollow

X-Robots-Tag: nofollow

noarchive

X-Robots-Tag: noarchive

nosnippet

X-Robots-Tag: nosnippet

max-snippet:[number]

X-Robots-Tag: max-snippet:150

max-video-preview:[number]

X-Robots-Tag: max-video-preview:30

noindex, nofollow

X-Robots-Tag: noindex, nofollow

HTTP Nginx add_header

nginx

```
location /path/to/page {  
    add_header X-Robots-Tag "noindex, nofollow";  
}
```

/path/to/page X-Robots-Tag: noindex, nofollow

Revision #2

Created 26 October 2024 08:54:38 by Ron

Updated 26 October 2024 09:06:24 by Ron