## 获取客户端并执行

git clone https://github.com/letsencrypt/letsencrypt

cd letsencrypt

./letsencrypt-auto --agree-dev-preview --server \

https://acme-v01.api.letsencrypt.org/directory auth

### 选择认证方式

在安装一些依赖包后，Let's Encrypt 将弹出 TUI 界面要求选择认证的方式：手动或独立。这里为了省事，选择独立认证。Standalone

接下来都选yes

## 接着输入 Email 地址

## 输入域名

## 在此，输入 bensonlin.me.org，多个域名使用逗号或空格分隔。

## 完成

当看到下列消息时，说明认证已经成功完成：

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at

/etc/letsencrypt/live/bensonlin.me/fullchain.pem. Your cert will

expire on 2017-05-23. To obtain a new or tweaked version of this

certificate in the future, simply run letsencrypt-auto again. To

non-interactively renew \*all\* of your certificates, run

"letsencrypt-auto renew"

- If you lose your account credentials, you can recover through

e-mails sent to 1096101803@qq.com.

- Your account credentials have been saved in your Certbot

configuration directory at /etc/letsencrypt. You should make a

secure backup of this folder now. This configuration directory will

also contain certificates and private keys obtained by Certbot so

making regular backups of this folder is ideal.

- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate

Donating to EFF: <https://eff.org/donate-le>

Let's Encrypt 将认证的信息保存于 /etc/letsencrypt 目录。

## 在 NGINX 的配置文件中将下面两行设置成 Let's Encrypt 的实际路径即可：

ssl\_certificate /etc/letsencrypt/live/bensonlin.me/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/ bensonlin.me /privkey.pem;

值得注意的是，目前 Let's Encrypt 的证书有效期为 90 天，之后需要手动续期。另外，在请求证书认证时会有频率限制。总的来说，证书的认证过程还是非常容易的，而且又是免费，所以对此有需要的朋友不妨一试。

Nginx配置

|  |
| --- |
| #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;  #log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  # '$status $body\_bytes\_sent "$http\_referer" '  # '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  #access\_log logs/access.log main;  sendfile on;  #tcp\_nopush on;  #keepalive\_timeout 0;  keepalive\_timeout 65;  #gzip on;  **#http重定向到https**  **server {**  **listen 80;**  **server\_name www.bensonlin.me;**    **rewrite ^(.\*)$ https://$host$1 permanent;**  **}**  server {  **listen 443;**  server\_name www.bensonlin.me;  **ssl on;**  **ssl\_certificate /etc/letsencrypt/live/bensonlin.me/fullchain.pem;**  **ssl\_certificate\_key /etc/letsencrypt/live/bensonlin.me/privkey.pem;**  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root D:/nginx-1.10.2/html/accountBook/public;  index index.php index.html index.htm;  try\_files $uri $uri/ /index.php?$query\_string;  }  #error\_page 404 /404.html;  # redirect server error pages to the static page /50x.html  #  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root /var/www/accountBook/resources/views/errors;  }  # proxy the PHP scripts to Apache listening on 127.0.0.1:80  #  #location ~ \.php$ {  # proxy\_pass http://127.0.0.1;  #}  # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000  #  location ~ \.php$ {  root /var/www/accountBook/public;  fastcgi\_pass 127.0.0.1:9000;  fastcgi\_index index.php;  fastcgi\_param SCRIPT\_FILENAME /var/www/accountBook/public$fastcgi\_script\_name;  include fastcgi\_params;  }  location ~ \.(gif|jpg|png|js|css|woff|ttf)$ {  root /var/www/accountBook/public;  }  # deny access to .htaccess files, if Apache's document root  # concurs with nginx's one  #  #location ~ /\.ht {  # deny all;  #}  }  # another virtual host using mix of IP-, name-, and port-based configuration  #  #server {  # listen 8000;  # listen somename:8080;  # server\_name somename alias another.alias;  # location / {  # root html;  # index index.html index.htm;  # }  #}  } |