**现在开始教程**

首先安装依存关系

#yum -y groupinstall "Development Tools"

#yum install vim git openssl-devel perl cpan make gcc g++

由于我们需要 Mojo::WebQQ 这个Perl语言项目作为主机端接收QQ消息的平台，同时CentOS 6.8自带yum源没有Cpanm包管理，所以接下来需要安装Cpanm

#cpan -i App::cpanminus

由于我们是第一次使用Cpan，程序会要求我们进行最基础的设定，一路回车即可，直到你看到

CPAN needs access to at least one CPAN mirror.

As you did not allow me to connect to the internet

you need to supply a valid CPAN URL now.

Please enter the URL of your CPAN mirror

不要惊慌，这是Cpan要求你设定一个Cpan仓库源，到这里找一个就可以，国内的主机可以去 China 列寻找，选择一个镜像服务器节点离你最近的地方，墙外的服务器可以直接用主站作为镜像源

Cpanm安装完毕后，我们就可以安装 Mojo::WebQQ 了

#cpanm Mojo::Webqq

如果你的手机不能直通国际互联网，一般情况下下载速度会非常慢或者完全无法安装，这时候你可以这样做

#cpanm --mirror http://mirrors.163.com/cpan/ Mojo::Webqq

如果出现了安装错误，一般情况下是 Mojolicious 未成功安装造成的，这时候你需要手动安装

#curl -L https://cpanmin.us | perl - -M https://cpan.metacpan.org -n Mojolicious

成功安装 Mojolicious 之后再次执行

#cpanm Mojo::Webqq

或

#cpanm --mirror http://mirrors.163.com/cpan/ Mojo::Webqq

应该就可以成功安装了

Mojo::Weixin的安装方法大致相同

**3.设定**

Mojo::WebQQ安装完毕后，就可以开始进行设定了

首先我们需要新建一个 Perl 脚本文件来执行

#touch GCM.pl

我们使用 vim 对文件进行操作

#vi GCM.pl

可能有人还不是很熟悉 vim 的使用方法，这里简单介绍一下会用到的功能

i进入编辑模式

ESC退出编辑模式

在非编辑模式下

:wq存盘退出vim

:q!不存盘退出vim

这里给出一个 Perl 脚本文件的模板，具体内容请根据实际情况替换更改

use Mojo::Webqq;

#微信使用use Mojo::Weixin

my $client = Mojo::Webqq->new(log\_encoding=>"utf-8");

$client->load("ShowMsg");

#请根据自己所需的推送服务进行选择并删除或注释不需要的部分，填写格式请仿照GCM的方式填写

#以下为GCM推送

$client->load("GCM",data=>{

api\_url => 'https://gcm-http.googleapis.com/gcm/send',

api\_key=>'AIzaSyB18io0hduB\_3uHxKD3XaebPCecug27ht8',

registration\_ids=>["输入你自己从GCMForMojo APP中获取到的令牌"],

allow\_group=>["接受群消息的号码，如需要推送全部群消息可删除这一行，每个群号码之间使用"",分隔"],

ban\_group=>[],

allow\_discuss=>[],

ban\_discuss=>[],

#此处为讨论组，填写格式同上

});

#以下为MiPush推送

$client->load("MiPush",data=>{

registration\_ids=>[""],

allow\_group=>[""],

ban\_group=>[],

allow\_discuss=>[],

ban\_discuss=>[],

});

#以下为HwPush推送

$client->load("HwPush",data=>{

registration\_ids=>[""],

allow\_group=>[""],

ban\_group=>[],

allow\_discuss=>[],

ban\_discuss=>[],

});

#以下为FmPush推送

$client->load("FmPush",data=>{

registration\_ids=>[""],

allow\_group=>[""],

ban\_group=>[],

allow\_discuss=>[],

ban\_discuss=>[],

});

$client->load("Openqq",data=>{

#如果是微信改为Openwx

listen => [{host=>"0.0.0.0",port=>5000}, ] ,

#如果是推送微信的话需要保证端口不重复，并请保证所设定的端口已经在防火墙内放行，同时需要在APP内设定好推送服务器的地址和端口

});

#不需要APP内回复功能请删除以上三行（不包括被#号注释掉的几行）

$client->run();

保存退出后执行

#perl GCM.pl

这时候你的 GCMForMojo APP应该会弹出一条检测到二维码事件的通知，点击它，使用手机端QQ扫描这个二维码，你的 GCMForMojo 就跑起来了

**注意事项**

Perl进程并不会后台运行！！！同时如果你的SSH连接中断的话当前终端下运行的全部会话均会被杀死，若想保持后台运行且断掉SSH连接后依旧可正常工作，请使用 screen 命令

#screen -S docker

#perl GCM.pl

然后请按 Ctrl+A ，再按 D 键使此 screen 进入后台驻守，然后就可以中断SSH连接了，如果需要恢复此 screen 的话，请执行

#screen -r docker

如果你没有能在 /storage/emulated/0/gcmformojo/ 里找到下载的二维码图片或接收到 GCMForMojo 的二维码事件通知，那么很可能你的 Mojo::WebQQ 版本过旧，如果需要升级 Mojo::WebQQ 的话，有两种方式

Cpanm仓库更新

#cpanm Mojo::Webqq

或

#cpanm --mirror http://mirrors.163.com/cpan/ Mojo::Webqq

Git仓库源编译安装升级

#yum install perl-ExtUtils-MakeMaker git

#git clone https://github.com/sjdy521/Mojo-Webqq.git

#cd Mojo-Webqq

#perl Makefile.PL

#make

#make install

#make clean

以后更新时

#cd Mojo-Webqq

#git pull

#perl Makefile.PL

#make

#make install

#make clean

Mojo::Weixin 的更新方式类似，请自行替换相应字符串

然后再执行

#perl GCM.pl

应该就可以看到二维码图片或接收到通知了

由于 OpenQQ 组件使用 HTTP 请求而不是更安全的 HTTPS 请求，这样会很容易被他人监听，并有可能以你的身份发送消息，所以为了安全起见，强烈建议以加盐或开启 HTTPS 加密的方式增强安全性

感谢Milkice同学整理增强安全性的方法，[原文链接在此](https://milkice.me/2017/03/gcmformojo-anquanxiangguanjiaocheng-security_tutorial/)

加盐

在 GCM.pl 文件内加入以下内容

use Mojo::Webqq;

use Digest::MD5 qw(md5 md5\_hex md5\_base64);

#请确保上一行加入在文件头行，否则会报错

#以下略

$client->load("Openqq",data=>{

listen => [{host=>"0.0.0.0",port=>5000}, ] ,

auth => sub {

my($param,$controller) = @\_;

my $secret = 'salt';

#请将该行salt改为你自定义的盐值，并在Android端内设定好你所自定义的盐值

my $text='';

foreach $key (sort keys %$param){

if($key ne 'sign'){

$value =$param->{$key};

$text.=$value;

}

}

if($param->{sign} eq md5\_hex($text.$secret) ){

return 1;

}

else{

return 0;

}

}

});

以下略

**加盐有什么用呢？**

盐（Salt），在密码学中，是指通过在密码任意固定位置插入特定的字符串，让散列后的结果和使用原始密码的散列结果不相符，这种过程称之为“加盐”。

（Via Wikipedia）

通俗来讲，这就是一个很简单的校验，如同一个密码，这样就可以很简单的解决之前提到的盗用问题，如果需要解决监听问题，你需要开启 HTTPS 加密

**HTTPS 加密**

在 GCM.pl 文件内加入以下内容

#以上略

$client->load("Openqq",data=>{

listen => [{

host =>"0.0.0.0",

port =>443,

#请求监听端口，默认443，也可以自定义

tls =>1,

#开启https请求支持

tls\_ca =>"/etc/tls/ca.crt"

#可选，ca证书路径

tls\_cert=>"/etc/tls/server.crt"

#服务器证书路径

tls\_key =>"/etc/tls/server.key"},],

#证书对应的key文件

});

HTTPS 证书可以使用 Let's Encrypt 这个免费的证书签发机构的证书，可信赖度不像之前 StartSSL 那么差，除了三个月需要续签一次以外没有什么大问题，签发和续期证书的步骤网上有很多，自行 Google 一下就好

服务端设定好后把 Android 端的服务器上将 http://xxxxxxxx:xxxx 改为 https://xxxxxxxx:xxxx 即可

注意，如果需要 HTTPS 加密，你需要申请一个域名并绑定在你的 推送服务器 上，否则你是无法签发可被信任的证书的，除非你选择了 自签发证书 ,但这样做会更麻烦

P.S.如果cpanm安装不成功，可以试试

发现yum安装perl无效，编译安装perl后安装cpanm包管理才行。

#wget http://www.cpan.org/src/5.0/perl-5.24.1.tar.gz

#tar -xzf perl-5.24.1.tar.gz

#cd perl-5.24.1

#./Configure -des -Dprefix=$HOME/localperl

#make

#make install

P.S.

通过HTTP代理连接Google FCM推送服务器

由于众所周知的原因，中国大陆无法直连FCM推送服务器。这种情况下为了能够向FFM客户端推送消息，可以在 config.js中的 push\_proxy 选项中指定FFM服务端连接Google FCM推送服务器时所使用的HTTP代理。具体思路请参考 [这个issue](https://github.com/RikkaApps/FCM-for-Mojo/issues/103) 。

客户端内回复功能:

https://www.ngrok.cc/#down-client一般手机(高通、联发科)选择Linux ARM版本,x86架构选Linux 32Bit版本

下载解压然后(sunny文件)上传到linux目录，这里我是扔到了root目录

注册并登陆: https://www.ngrok.cc

购买开通隧道,端口为配置文件设置的(例如我的5000)

[https://img.ithome.com/images/v2/t.png](http://img.ithome.com/newsuploadfiles/2018/2/20180214_225438_355.png)[https://img.ithome.com/images/v2/t.png](http://img.ithome.com/newsuploadfiles/2018/2/20180214_225439_403.png)

[https://img.ithome.com/images/v2/t.png](http://img.ithome.com/newsuploadfiles/2018/2/20180214_225438_857.png)

复制隧道id粘贴替换到下文

Ssh连接Linux逐条运行下面命令

#screen -S n

#cd /root

#chmod a+x sunny

#./sunny clientid +你的隧道ID

在转发一行出现你当时注册的网址时就好了

把网址复制到GCM客户端即可回复消息。

文中所提到的工具[下载链接](http://yzs.srcol.com/tools.zip)

P.S.本文中涉及到的引用均遵循知识共享署名-非商业性使用-相同方式共享4.0国际许可协议

其中涉及到的原作品地址以及作者如下

作者kotomei

链接：https://gist.github.com/kotomei/5367a003cd16d05e075c21a7f360b09a

作者Rikka，原作品链接：https://github.com/RikkaApps/FCM-for-Mojo/blob/master/README.md

其中第一篇作者在文章中注明按照该协议进行许可，第二篇作者回复按照该协议进行许可