

版本历史修订记录

版本号	日期	修改状态	修改内容	编辑人
V 1.0	2019/3	创建	创建文档	亚可

项目简介

这个项目是针对业务的一个聊天机器人的聊天框架，集成了 qqbot、wxchat、rasa 以及 web 服务。提供微信和 qq 聊天接口，可以对某个群组发送系统监控消息等。QQ 和微信聊天机器人都无法直接发消息给群组中的非好友用户，只能通过 @ 的方式提醒。

[微信聊天机器人](http://wxpy.readthedocs.io/zh/latest/) 使用的是开源的[wspy](<http://wxpy.readthedocs.io/zh/latest/>) 机器人框架。使用 @register 这种 python wapper 机制来绑定消息响应函数。微信机器人需要对群组进行注册才能对消息进行发送。

[QQ 聊天机器人](#) 采用酷 Q 方案。酷 Q 是 windows 下的一个程序，可以模拟 qq。在 linux 系统中，采用 wine 技术。需要单独部署成为一个 docker，然后通过一个 http 的接口暴露，各类 http 接口，用来让我们发送消息。提供 http 消息回调。

全局目录结构

Main.py 系统的主入口

- | -bin/ 启动、停止脚本
- | -bot/ bot机器人相关代码
 - | | -wxpy 微信机器人
 - | | -qqbot qq机器人
 - | | -rasa 对话机器人
- | -data/ 数据库初始脚本和其他预制数据
- | -business/ 业务处理器，对应各类业务的处理类
- | -common/ 一些共通处理，基础类
- | -database/ 单独的一个数据库中间件服务器，用于部署于app区做数据库代理
- | -log/ 用于存放日志文件
- | -test/ 测试相关的脚本
- | -web/ Web服务器的程序

实现功能

1. 接受内部系统，如监控系统的系统调用，从而把消息推送给 QQ 或者微信用户

内部系统调用服务的时候，需要提供以下信息

- 发给哪个群组
- 发给这个群组中的那个用户
- 发送的消息

2. 可以接受 QQ、微信用户的对话，理解其意图，并且回应用户

项目部署&安装

此项目主要针对 linux 系统，采用 python 2.7+环境进行部署说明。

依赖环境安装

1、安装必需依赖包：

取项目中的 ns4_chatbot/requirements.txt 文件，运行

pip install -r requirements.txt 命令

2、安装 redis

3、安装 mysql 数据库，导入 ns4_chatbot/import_data/production/ddl.sql 文件

酷 Q 聊天机器人安装

酷 Q 目前可以在 Wine 中运行，见 酷 Q Air / Pro on Wine，因此也就自然而然有了相应的 Docker 镜像 coolq/wine-coolq。

要在 Docker 中使用本插件，你可以使用酷 Q 官方的 Docker 镜像，然后在其中安装本插件（下载 cpk、编辑配置文件、启用插件），也可以使用维护的已安装并启用了插件的镜像 richardchien/cqhttp（基于酷 Q 官方的镜像修改）。下面介绍这个镜像的用法。

1、docker 安装

```
sudo apt-get install -y docker.io
```

2、酷 Q 安装

```
$ docker pull richardchien/cqhttp:latest
$ mkdir coolq # 用于存储酷 Q 的程序文件
$ docker run -ti --rm --name cqhttp-test \
    # 将宿主目录挂载到容器内用于持久化酷 Q 的程序文件
    -v $(pwd)/coolq:/home/user/coolq \
    -p 9000:9000 \ # noVNC 端口，用于从浏览器控制酷 Q
    -p 5700:5700 \ # HTTP API 插件开放的端口
    -e COOLQ_ACCOUNT=123456 \ # 要登录的 QQ 账号，可选但建议填
    -e CQHTTP_POST_URL=http://example.com:8080 \ # 事件上报地址
```

```
# 允许通过 HTTP 接口访问酷 Q 数据文件  
-e CQHTTP_SERVE_DATA_FILES=yes \  
richardchien/cqhttp:latest
```

3、配置酷 Q 回调地址：

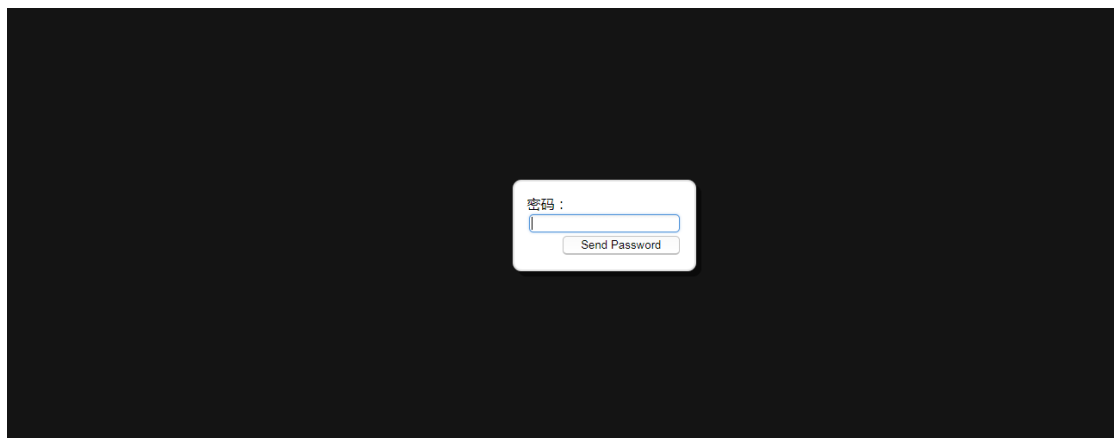
在 `/coolq/app/io.github.richardchien.coolqhttpapi/config/<QQ_ID>.ini`
中添加 `post_url=http://[IP]:8080/coolq_callback`

4、酷 Q 启动：

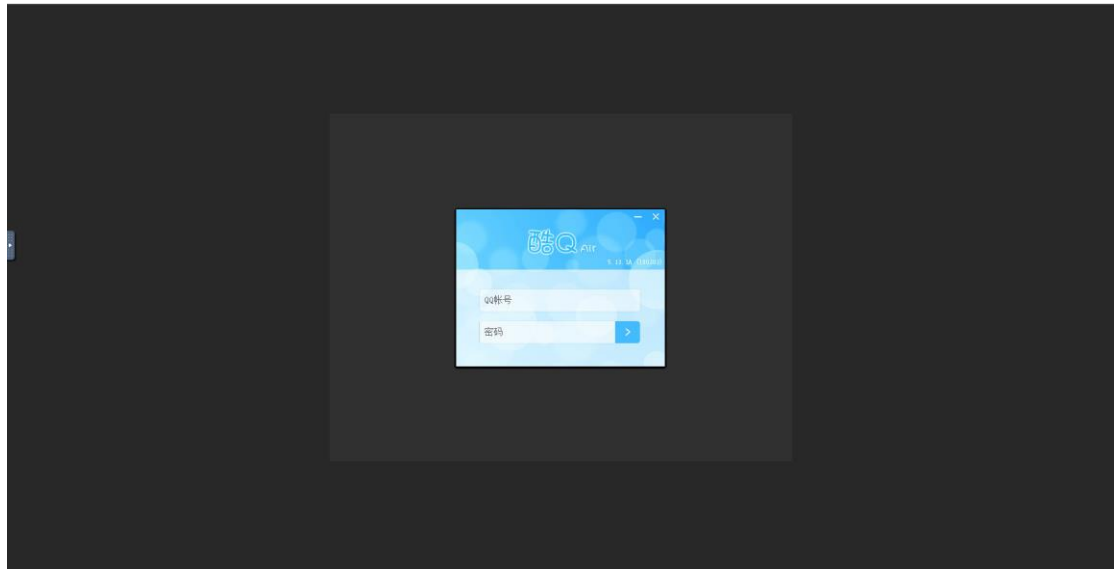
1、在浏览器中输入 [http://\[ip\]:5700](http://[ip]:5700)，进入如下页面：



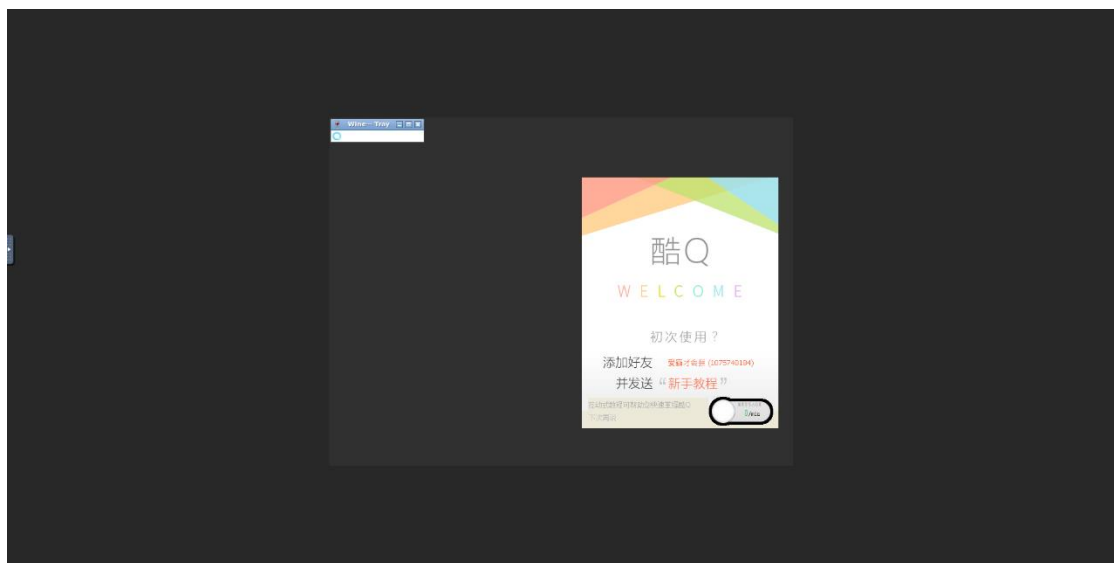
2、点击图中的链接，进入如下页面，密码为 COOLQ_ACCOUNT 对应的密码



3、登录酷 Q，输入 qq 密码等



4、登录成功界面



5、查看日志：



项目配置&启动

1、配置 bot.conf 文件

```
bot=wechat,qq #启动微信和 QQ 功能

bot_chat_base_inteval = 1 #设置消息的最少发送间隔，防止微信和 QQ 被封

bot_chat_interval = 3 #在 bot_chat_base_inteval 的基础上随机停顿的时间

retry_max_num = 10 #微信或者 QQ 掉线后尝试重新登录的次数

retry_interval = 60 #尝试重新登录的时间间隔

retry_sleep = 600 #尝试 10 次后，没有登录成功，就彻底休息 10 分钟

debug = True #是否打开调试日志

[admin] email #微信二维码和错误日志发送的邮箱地址，支持多个邮箱，多个邮箱地址以逗号隔开

[email]模块 #配置发送邮箱的用户名密码等，注意，此邮箱需要开通 smtp 协议

[http]模块

    port = 8080 #配置 http 服务监听地址端口

[db] 模块 #配置数据库的用户名、密码等相关信息

[redis] 模块 #配置 redis 的 ip、端口和密码等信息

[wxbot]模块

    console_qr = True #是否在日志窗口打印微信登录二维码
```

```
cache_path = .cache #缓存地址，存放系统产生的图片，二维码等信息
cache_file = wxpy.pkl #存放微信记住密码等文件
qr_path = .cache/qr.png #存放微信二维码

[coolq] 模块 #配置酷Q 相关的信息

url=http://127.0.0.1:5700/ #配置登录酷Q 的地址
qq=12345678 #配置酷Q 的登录账号

cache_path = [workspace]/coolqdata/image #配置酷Q 的图片位置，必须配置到 coolq 的运行目录的 data/image，这样 CQ 码发送的时候图片的时候才能生效

[voice2txt] ##科大讯飞|百度的声音转文本配置

[rasa]

#NLU 自然语言意图识别模型

nlu_model = rasa/model/default/latest

#对话模型

dialog_model = rasa/model/dialogue
```

2、数据库配置

在数据库 biz_system_tree 中定义所有可能用到的 qq 群组 ID 和微信群组名称，如果不定义的话，可能会出现找不到对应的群组的错误

3、启动

在 ns4_chatbot 目录下面，运行 nohup python Main.py 进行项目的启动。然后到配置的邮箱进行微信扫码。

接口调用说明

发送消息接口地址【post】：<https://<ip>:<port>/chat>

请求参数：

参数定义	参数名称	是否必须	取值取值示范	说明
qqGroupId	qq 群组 ID	否	22263743	

wxGroupName	微信群组名称	否	自动化组	
msgId	消息 ID	是	2019031100001	
type	标识发送微信	是	wechat	qq wechat
user	要@的用户	否	张三	
msg	文本消息	否	监控报警	
html	Html 格式，会自动转换为图片格式发送给用户	否	<html><head></head><body>监控报警</body></html>	
remark	备注	否		