python 实现微信接口(itchat)

安装

- `sudo pip install itchat`

登录

- `itchat.auto_login()` 这种方法将会通过微信扫描二维码登录，但是这种登录的方式确实短时间的登录，并不会保留登录的状态，也就是下次登录时还是需要扫描二维码，如果加上 `hotReload==True`,那么就会保留登录的状态，至少在后面的几次登录过程中不会再次扫描二维码，该参数生成一个静态文件 `itchat.pkl` 用于存储登录状态

退出及登录完成后调用的特定的方法

这里主要使用的是灰调函数的方法,登录完成后的方法需要赋值

在 `loginCallback` 中退出后的方法,需要赋值在 `exitCallback` 中.若不设置 `loginCallback` 的值，将会自动删除二维码图片并清空命令行显示.

```
import itchat, time

def lc():

    print("Finash Login!")

def ec():
```

```
    print("exit")


itchat.auto_login(loginCallback=lc, exitCallback=ec)

time.sleep()

itchat.logout()     #强制退出登录
```

回复消息

---

send

- `send(msg="Text Message", toUserName=None)`

参数：

- `msg`：文本消息内容
- `@fil@path_to_file`：发送文件
- `@img@path_to_img`：发送图片
- `@vid@path_to_video`：发送视频
- `toUserName`：发送对象，如果留空，将发送给自己.

返回值

- `True` or `False`

实例代码

```
# coding-utf-8

import itchat
```

```
itchat.auto_login()

itchat.send("Hello World!")

ithcat.send("@fil@%s" % '/tmp/test.text')

ithcat.send("@img@%s" % '/tmp/test.png')

ithcat.send("@vid@%s" % '/tmp/test.mkv')
```

send_msg

- `send_msg(msg='Text Message', toUserName=None)`,其中的的 `msg` 是要发送的文本，`toUserName` 是发送对象，如果留空，将发送给自己，返回值为 `True` 或者 `False`

实例代码

```
import itchat

itchat.auto_login()

itchat.send_msg("hello world.")
```

send_file

- `send_file(fileDir, toUserName=None)` `fileDir` 是文件路径，当文件不存在时，将打印无此文件的提醒，返回值为 `True` 或者 `False`

实例代码

```
mport itchat

itchat.auto_login()

itchat.send_file("/tmp/test.txt")
```

send_image

- `send_image(fileDir, toUserName=None)` 参数同上

实例代码

```
import itchat

itchat.auto_login()

itchat.send_img("/tmp/test.txt")
```

send_video

- `send_video(fileDir, toUserName=None)` 参数同上

实例代码

```
import itchat



itchat.auto_login()

itchat.send_video("/tmp/test.txt")
```

注册消息方法

`itchat` 将根据接受到的消息类型寻找对应的已注册的方法.

如果一个消息类型没有对应的注册方法，该消息将会被舍弃.

在运行过程中也可以动态注册方法，注册方式与结果不变.

注册方法

- 不带具体对象注册，将注册为普通消息的回复方法.

```
import itchat

from itchat.content import *

@itchat.msg_register(TEXT)   #这里的 TEXT 表示如果有人发送文本消息，那么就会调用下面的方法

def simple_reply(msg):

    #这个是向发送者发送消息

    itchat.send_msg('已经收到了文本消息，消息内容为%s'%msg['Text'],toUserName=msg['FromUserName'])

    return "T reveived: %s" % msg["Text"]      #返回的给对方的消息，msg["Text"]表示消息的内容
```

- 带对象参数注册，对应消息对象将调用该方法，其中 `isFriendChat` 表示好友之间，`isGroupChat` 表示群聊，`isMapChat` 表示公众号

```
import itchat

from itchat.content import *


@itchat.msg_register(TEXT, isFriendChat=True, isGroupChat=True,isMpChat=True)

def text_reply(msg):

    msg.user.send("%s : %s" % (mst.type, msg.text))
```

- 消息类型

  向注册方法传入的 `msg` 包含微信返回的字典的所有内容.`itchat` 增加 `Text`, `Type`(也就是参数) 键值，方便操作.

`itcaht.content` **中包含所有的消息类型参数，如下表**

参数	**l 类型**	**Text 键值**
TEXT	文本	文本内容(文字消息)
MAP	地图	位置文本(位置分享)
CARD	名片	推荐人字典(推荐人的名片)
SHARING	分享	分享名称(分享的音乐或者文章等)
PICTURE 下载方法		图片/表情
RECORDING	语音	下载方法
ATTACHMENT	附件	下载方法

参数	l 类型	Text 键值
VIDEO	小视频	下载方法
FRIENDS	好友邀请	添加好友所需参数
SYSTEM	系统消息	更新内容的用户或群聊的 UserName 组成的列表
NOTE	通知	通知文本(消息撤回等)

附件的下载与发送

itchat 的附件下载方法存储在 msg 的 Text 键中.

发送的文件名(图片给出的默认文件名), 都存储在 msg 的 FileName 键中.

下载方法, 接受一个可用的位置参数(包括文件名), 并将文件响应的存储.

注意：下载的文件存储在指定的文件中，直接将路径与 FileName 连接即可，如 msg["Text"]('/tmp/weichat'+msg['FileName'])

```
@itchat.msg_register([PICTURE, RECORDING, ATTACHMENT, VIDEO])

def download_files(msg):

    #msg.download(msg['FileName'])    #这个同样是下载文件的方式

    msg['Text'](msg['FileName'])       #下载文件
```

```
    #将下载的文件发送给发送者

    itchat.send('@%s@%s' % ('img' if msg['Type'] == 'Picture' else 'fil', msg["FileName"]), msg["FromUserName"])
```

群消息

增加了三个键值，如下：

- `isAt` 判断是否 @ 本号
- `ActualNickName`：实际 `NickName`(昵称)
- `Content`：实际 `Content`

测试程序

```
import itcaht

from itchat.content import TEXT



@itchat.msg_register(TEXT, isGroupChat=True)

def text_reply(msg):

    if(msg.isAt):    #判断是否有人@自己

    #如果有人@自己，就发一个消息告诉对方我已经收到了信息

    itchat.send_msg("我已经收到了来自{0}的消息，实际内容为{1}".format(msg['ActualNickName'],msg['Text']),toUserName=msg['FromUserName'])



itchat.auto_login()

itchat.run()
```

注册消息的优先级

总的来说就是后面注册同种类型的消息会覆盖之前注册的消息，详情见文档

https://itchat.readthedocs.io/zh/latest/

消息内容

注意：所有的消息内容都是可以用键值对来访问的，如 `msg["FromUserName]`就是查看发送者，

`itchat.search_friends(userName=msg['FromUserName'])['NickName']`查看的是当发送者昵称

一般消息

一般的消息都遵循以下的内容：

```
{
    "FromUserName": "",
    "ToUserName": "",
    "Content": "",
    "StatusNotifyUserName": "",
    "ImgWidth": 0,
    "PlayLength": 0,
    "RecommendInfo": {},
    "StatusNotifyCode": 0,
```

```
    "NewMsgId": "",
    "Status": 0,
    "VoiceLength": 0,
    "ForwardFlag": 0,
    "AppMsgType": 0,
    "Ticket": "",
    "AppInfo": {},
    "Url": "",
    "ImgStatus": 0,
    "MsgType": 0,
    "ImgHeight": 0,
    "MediaId": "",
    "MsgId": "",
    "FileName": "",
    "HasProductId": 0,
    "FileSize": "",
    "CreateTime": 0,
    "SubMsgType": 0
}
```


初始化消息

MsgType: 51

FromUserName: 自己 ID

ToUserName: 自己 ID

StatusNotifyUserName: 最近联系的联系人 ID

Content:

```
    <msg>
        <op id='4'>
            <username>
                # 最近联系的联系人
                filehelper,xxx@chatroom,wxid_xxx,xxx,...
            </username>
            <unreadchatlist>
                <chat>
                    <username>
                    # 朋友圈
                        MomentsUnreadMsgStatus
                    </username>
                    <lastreadtime>
                        1454502365
                    </lastreadtime>
                </chat>
            </unreadchatlist>
```

```
            <unreadfunctionlist>
            # 未读的功能账号消息，群发助手，漂流瓶等
            </unreadfunctionlist>
        </op>
    </msg>
```

文本消息

MsgType: 1

FromUserName: 发送方 ID

ToUserName: 接收方 ID

Content: 消息内容

图片消息

itchat 增加了 Text 键，键值为 下载该图片的方法.

MsgType: 3

FromUserName: 发送方 ID

ToUserName: 接收方 ID

MsgId: 用于获取图片，用于表示每一条消息

Content:

```
    <msg>
        <img length="6503" hdlength="0" />
        <commenturl></commenturl>
    </msg>
```

拓展：如果想要得到 `Content` 中的具体内容可以使用正则表达式匹配出来

视频消息

`itchat` 增加了 `Text` 键，键值为 下载该视频的方法.

```
    MsgType: 62

    FromUserName: 发送方 ID

    ToUserName: 接收方 ID

    MsgId: 用于获取小视频

    Content:

        <msg>

            <img length="6503" hdlength="0" />

            <commenturl></commenturl>

        </msg>
```

地理位置消息

`itchat` 增加了 `Text` 键，键值为 该地点的文本形式.

```
MsgType: 1

    FromUserName: 发送方 ID

    ToUserName: 接收方 ID

    Content: http://weixin.qq.com/cgi-bin/redirectforward?args=xxx



    OriContent:<?xml version="1.0"?>

<msg>
```

```
    <location x="34.195278" y="117.177803" scale="16" label="江苏省徐州市铜山区新区海河路" maptype="0" poiname="江苏师范大学大学生公寓园区" />

</msg>
```

名片消息

itchat 增加了 Text 键，键值为 该调用 add_friend 需要的属性.

```
 MsgType: 42

    FromUserName: 发送方 ID

    ToUserName: 接收方 ID

    Content:

        <?xml version="1.0"?>

        <msg bigheadimgurl="" smallheadimgurl="" username="" nickname="" shortpy="" alias="" imagestatus="3" scene="17" province="" city="" sign="" sex="1" certflag="0" certinfo="" brandIconUrl="" brandHomeUrl="" brandSubscriptConfigUrl="" brandFlags="0" regionCode="" />



    RecommendInfo:

        {

            "UserName": "xxx", # ID，这里的是昵称

            "Province": "xxx",

            "City": "xxx",

            "Scene": 17,

            "QQNum": 0,

            "Content": "",
```

```
            "Alias": "xxx", # 微信号
            "OpCode": 0,
            "Signature": "",
            "Ticket": "",
            "Sex": 0, # 1:男, 2:女
            "NickName": "xxx", # 昵称
            "AttrStatus": 4293221,
            "VerifyFlag": 0
        }
```

下面是添加好友的测试代码

```
@itchat.msg_register(itchat.content.CARD,isFriendChat=True)
def simply(msg):
    print msg['Text']
    print msg['Content']
    itchat.add_friend(userName=msg['Text']['UserName'])  #添加推荐的好友
    print msg['RecommendInfo']
    print msg['RecommendInfo']['UserName']
```

语音消息

`itchat` 增加了 `Text` 键,键值为下载该语音文件的方法,下载下来的是 `MP3` 的格式

```
MsgType: 34
```

FromUserName: 发送方 ID

ToUserName: 接收方 ID

MsgId: 用于获取语音

Content:

<msg>

<voicemsg endflag="1" cancelflag="0" forwardflag="0" voiceformat="4" voicelength="1580" length="2026" bufid="216825389722501519" clientmsgid="49efec63a9774a65a932a4e5fcd4e923filehelper174_1454602489" fromusername="" />

</msg>

下载方法：`msg['Text'](msg['FileName'])`

动画表情

`itchat` 添加了 `Text` 键，键值为下载该图片表情的方法。

注意：本人亲测对于一些微信商店提供的表情是不能下载成功的,这里的自带的表情 `emoji` 是属于 `TEXT` 类别的，因此如果将其注册为 `PICTURE` 消息类型的话是不可以监测到的

MsgType: 47

FromUserName: 发送方 ID

ToUserName: 接收方 ID

Content:

<msg>

<emoji fromusername = "" tousername = "" type="2" idbuffer="media:0_0" md5="e68363487d8f0519c4e1047de403b2e7" len = "86235" pro

```
ductid="com.tencent.xin.emoticon.bilibili" androidmd5="e68363487d8f0519c4e1047de403b2e7" androidlen="86235" s60v3md5 = "e68363487d8f0519c4e1047de403b2e7" s60v3len="86235" s60v5md5 = "e68363487d8f0519c4e1047de403b2e7" s60v5len="86235" cdnurl = "http://emoji.qpic.cn/wx_emoji/eFygWtxcoMF8M0oCCsksMA0gplXAFQNpiaqsmOicbXl1OC4Tyx18SGsQ/" designerid = "" thumburl = "http://mmbiz.qpic.cn/mmemoticon/dx4Y70y9XctRJf6tKsy7FwWosxd4DAtItSfhKS0Czr56A70p8U5O8g/0" encrypturl = "http://emoji.qpic.cn/wx_emoji/UyYVK8GMlq5VnJ56a4GkKHAiaC266Y0me0KtW6JN2FAZcXiaFKccRevA/" aeskey= "a911cc2ec96ddb781b5ca85d24143642" ></emoji>

            <gameext type="0" content="0" ></gameext>

        </msg>
```

普通链接或应用分享消息

主要针对的是分享的文章等等

```
    MsgType: 49

    AppMsgType: 5

    FromUserName: 发送方 ID

    ToUserName: 接收方 ID

    Url: 链接地址

    FileName: 链接标题

    Content:

        <msg>

            <appmsg appid=""  sdkver="0">

                <title></title>

                <des></des>

                <type>5</type>
```

```
            <content></content>
            <url></url>
            <thumburl></thumburl>
            ...
        </appmsg>
        <appinfo>
            <version></version>
            <appname></appname>
        </appinfo>
    </msg>
```

音乐链接消息

主要针对的是音乐

```
MsgType: 49
    AppMsgType: 3
    FromUserName: 发送方 ID
    ToUserName: 接收方 ID
    Url: 链接地址
    FileName: 音乐名

    AppInfo: # 分享链接的应用
```

```
    {
        Type: 0,
        AppID: wx485a97c844086dc9
    }

Content:
    <msg>
        <appmsg appid="wx485a97c844086dc9"  sdkver="0">
            <title></title>
            <des></des>
            <action></action>
            <type>3</type>
            <showtype>0</showtype>
            <mediatagname></mediatagname>
            <messageext></messageext>
            <messageaction></messageaction>
            <content></content>
            <contentattr>0</contentattr>
            <url></url>
            <lowurl></lowurl>
            <dataurl>
```

```
                http://ws.stream.qqmusic.qq.com/C100003i9hMt1bgui
0.m4a?vkey=6867EF99F3684&amp;guid=ffffffffc104ea2964a111cf3ff3edaf&a
mp;fromtag=46

            </dataurl>

            <lowdataurl>

                http://ws.stream.qqmusic.qq.com/C100003i9hMt1bgui
0.m4a?vkey=6867EF99F3684&amp;guid=ffffffffc104ea2964a111cf3ff3edaf&a
mp;fromtag=46

            </lowdataurl>

            <appattach>

                <totallen>0</totallen>

                <attachid></attachid>

                <emoticonmd5></emoticonmd5>

                <fileext></fileext>

            </appattach>

            <extinfo></extinfo>

            <sourceusername></sourceusername>

            <sourcedisplayname></sourcedisplayname>

            <commenturl></commenturl>

            <thumburl>

                http://imgcache.qq.com/music/photo/album/63/180_al
bumpic_143163_0.jpg

            </thumburl>

            <md5></md5>
```

```
            </appmsg>

            <fromusername></fromusername>

            <scene>0</scene>

            <appinfo>

                <version>29</version>

                <appname>摇一摇搜歌</appname>

            </appinfo>

            <commenturl></commenturl>

        </msg>
```

群消息

itchat 增加了三个群聊相关的键值:

- isAt：判断是否 @ 本号
- ActualNickName：实际 NickName
- Content：实际 Content

```
MsgType: 1

FromUserName: @@xxx

ToUserName: @xxx

Content:

    @xxx:<br/>xxx
```

红包消息

```
 MsgType: 49
```

AppMsgType: 2001

FromUserName: 发送方 ID

ToUserName: 接收方 ID

Content: 未知

系统消息

MsgType: 10000

FromUserName: 发送方 ID

ToUserName: 自己 ID

Content:

"你已添加了 xxx ，现在可以开始聊天了。"

"如果陌生人主动添加你为朋友，请谨慎核实对方身份。"

"收到红包，请在手机上查看"

账号类型

`tchat` 为三种账号都提供了 整体获取方法与搜索方法.

好友

get_friends

- `itchat.get_friends()` 返回完整的好友列表
- 每个好友为一个字典, 其中第一项为本人的账号信息;
- 传入 `update=True`, 将更新好友列表并返回, `get_friends(update=True)`

search_friends

- itchat.get_friends() 好友搜索，有以下四种方式
- 仅获取自己的用户信息

```
# 获取自己的用户信息，返回自己的属性字典

itchat.search_friends()
```

- 获取特定 `UserName` 的用户信息

```
# 获取特定 UserName 的用户信息

itchat.search_friends(userName='@abcdefg1234567')



## 获取发送信息的好友的详细信息

@itchat.msg_register(itchat.content.TEXT,isFriendChat=True)

def reply(msg):

    print msg['FromUserName']

    print itchat.search_friends(userName=msg['FromUserName'])    #详细信息

    print itchat.search_friends(userName=msg['FromUserName'])['NickName']    #获取昵称
```

- 获取备注,微信号，昵称中的任何一项等于 `name` 键值的用户. (可以与下一项配置使用.)

比如在我的微信中有一个备注为 `autolife` 的人，我可以使用这个方法搜索出详细的信息

```
# 获取任何一项等于 name 键值的用户

itchat.search_friends(name='autolife')
```

- 获取备注,微信号，昵称分别等于相应键值的用户. (可以与上一项配置使用.)

```
# 获取分别对应相应键值的用户

itchat.search_friends(wechatAccount='littlecodersh')

# 三、四项功能可以一同使用

itchat.search_friends(name='LittleCoder 机器人', wechatAccount='littlecodersh')
```

update_friend

主要用于好友更新

- 特定用户: 传入用户 `UserName`，返回指定用户的最新信息.
- 用户列表: 传入 `UserName` 组成的列表，返回用户最新信息组成的列表

```
memberList = itchat.update_friend('@abcdefg1234567')
```

公众号

get_mps

将返回完整的工作号列表

- 每个公众号为一个字典,

- 传入 `update=True` 将更新公众号列表，并返回.

search_mps

- 获取特定 `UserName` 的公众号

```
# 获取特定 UserName 的公众号，返回值为一个字典

itchat.search_mps(userName='@abcdefg1234567')
```

- 获取名字中还有特定字符的公众号.

```
# 获取名字中含有特定字符的公众号，返回值为一个字典的列表

itchat.search_mps(name='LittleCoder')
```

- 当两项都是勇士，将仅返回特定 `UserName` 的公众号.

群聊

- `get_chatrooms`：返回完整的群聊列表.
- `search_chatrooms`：群聊搜索.
- `update_chatroom`：获取群聊用户列表或更新该群聊.
- 群聊在首次获取中不会获取群聊的用户列表，所以需要调用该命令才能获取群聊成员.
- 传入群聊的 `UserName`，返回特定群聊的详细信息.
- 传入 `UserName` 组成的列表，返回指定用户的最新信息组成的列表.

```
memberList = itchat.update_chatroom('@@abcdefg1234567', detailedMember=True)
```

- 创建群聊,增加/删除群聊用户:

- 由于之前通过群聊检测是否被好友拉黑的程序，目前这三个方法都被严格限制了使用频率.
- 删除群聊需要本账号为管理员，否则无效.
- 将用户加入群聊有直接加入与发送邀请，通过 `useInvitation` 设置.
- 超过 40 人的群聊无法使用直接加入的加入方式.

```
memberList = itchat.get_frients()[1:]

# 创建群聊，topic 键值为群聊名称.

chatroomUserName = itchat.create_chatroom(memberList, "test chatroom
")

# 删除群聊内的用户

itchat.delete_member_from_chatroom(chatroomUserName, memberList[0])

# 增加用户进入群聊.

itchat.add_member_into_chatroom(chatroomUserName, memberList[0], use
Invitation=False)
```

方法汇总

```
itchat.add_friend

itchat.new_instance

itchat.add_member_into_chatroom

itchat.originInstance

itchat.auto_login

itchat.returnvalues

itchat.check_login
```

itchat.run

itchat.components

itchat.search_chatrooms

itchat.config

itchat.search_friends

itchat.configured_reply

itchat.search_mps

itchat.content

itchat.send

itchat.core

itchat.send_file

itchat.Core

itchat.send_image

itchat.create_chatroom

itchat.send_msg

itchat.delete_member_from_chatroom

itchat.send_raw_msg

itchat.dump_login_status

itchat.send_video

itchat.get_chatrooms

itchat.set_alias

itchat.get_contact

itchat.set_chatroom_name

itchat.get_friends

itchat.set_logging

itchat.get_head_img

itchat.set_pinned

itchat.get_mps

itchat.show_mobile_login

itchat.get_msg

itchat.start_receiving

itchat.get_QR

itchat.storage

itchat.get_QRuuid

itchat.update_chatroom

itchat.instanceList

itchat.update_friend

itchat.load_login_status

itchat.upload_file

itchat.log

itchat.utils

itchat.login

```
itchat.VERSION

itchat.logout

itchat.web_init

itchat.msg_register
```

实例

下面是博主写的一个程序，该程序的主要功能是监控撤回消息，并且如果有消息撤回就会撤回的消息发送给你，以后再也不用担心看不到好友的撤回的消息了，由于注释写的很详细，因此这里就不在详细的讲解了，直接贴代码

代码

```
# coding:utf-8

import itchat

from itchat.content import TEXT

from itchat.content import *

import sys

import time

import re



reload(sys)

sys.setdefaultencoding('utf8')

import os
```

```
msg_information = {}

face_bug=None  #针对表情包的内容




@itchat.msg_register([TEXT, PICTURE, FRIENDS, CARD, MAP, SHARING, REC
ORDING, ATTACHMENT, VIDEO],isFriendChat=True, isGroupChat=True, isMpC
hat=True)

def handle_receive_msg(msg):

    global face_bug

    msg_time_rec = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime
())   #接受消息的时间

    msg_from = itchat.search_friends(userName=msg['FromUserName'])['N
ickName']   #在好友列表中查询发送信息的好友昵称

    msg_time = msg['CreateTime']    #信息发送的时间

    msg_id = msg['MsgId']    #每条信息的 id

    msg_content = None      #储存信息的内容

    msg_share_url = None    #储存分享的链接，比如分享的文章和音乐

    print msg['Type']

    print msg['MsgId']

    if msg['Type'] == 'Text' or msg['Type'] == 'Friends':     #如果发
送的消息是文本或者好友推荐

        msg_content = msg['Text']

        print msg_content
```

```
    #如果发送的消息是附件、视屏、图片、语音
    elif msg['Type'] == "Attachment" or msg['Type'] == "Video" \
            or msg['Type'] == 'Picture' \
            or msg['Type'] == 'Recording':
        msg_content = msg['FileName']    #内容就是他们的文件名
        msg['Text'](str(msg_content))    #下载文件
        # print msg_content
    elif msg['Type'] == 'Card':    #如果消息是推荐的名片
        msg_content = msg['RecommendInfo']['NickName'] + '的名片'    #
内容就是推荐人的昵称和性别
        if msg['RecommendInfo']['Sex'] == 1:
            msg_content += '性别为男'
        else:
            msg_content += '性别为女'


        print msg_content
    elif msg['Type'] == 'Map':    #如果消息为分享的位置信息
        x, y, location = re.search(
            "<location x=\"(.*?)\" y=\"(.*?)\".*label=\"(.*?)\".*", ms
g['OriContent']).group(1, 2, 3)
        if location is None:
```

```
            msg_content = r"纬度->" + x.__str__() + " 经度->" + y.__str
__()      #内容为详细的地址

        else:

            msg_content = r"" + location

    elif msg['Type'] == 'Sharing':      #如果消息为分享的音乐或者文章，详
细的内容为文章的标题或者是分享的名字

        msg_content = msg['Text']

        msg_share_url = msg['Url']       #记录分享的 url

        print msg_share_url

    face_bug=msg_content



##将信息存储在字典中，每一个 msg_id 对应一条信息

    msg_information.update(

        {

            msg_id: {

                "msg_from": msg_from, "msg_time": msg_time, "msg_time_
rec": msg_time_rec,

                "msg_type": msg["Type"],

                "msg_content": msg_content, "msg_share_url": msg_share
_url

            }

        }

    )
```

```
##这个是用于监听是否有消息撤回

@itchat.msg_register(NOTE, isFriendChat=True, isGroupChat=True, isMp
Chat=True)

def information(msg):

    #这里如果这里的 msg['Content']中包含消息撤回和 id，就执行下面的语句

    if '撤回了一条消息' in msg['Content']:

        old_msg_id = re.search("\<msgid\>(.*?)\<\/msgid\>", msg['Cont
ent']).group(1)   #在返回的 content 查找撤回的消息的 id

        old_msg = msg_information.get(old_msg_id)    #得到消息

        print old_msg

        if len(old_msg_id)<11:  #如果发送的是表情包

            itchat.send_file(face_bug,toUserName='filehelper')

        else:  #发送撤回的提示给文件助手

            msg_body = "告诉你一个秘密~" + "\n" \

                       + old_msg.get('msg_from') + " 撤回了 " + old_ms
g.get("msg_type") + " 消息" + "\n" \

                       + old_msg.get('msg_time_rec') + "\n" \

                       + "撤回了什么 ⇣" + "\n" \

                       + r"" + old_msg.get('msg_content')

            #如果是分享的文件被撤回了，那么就将分享的 url 加在 msg_body 中发
送给文件助手
```

```
            if old_msg['msg_type'] == "Sharing":
                msg_body += "\n就是这个链接➢ " + old_msg.get('msg_shar
e_url')


            # 将撤回消息发送到文件助手
            itchat.send_msg(msg_body, toUserName='filehelper')
            # 有文件的话也要将文件发送回去
            if old_msg["msg_type"] == "Picture" \
                    or old_msg["msg_type"] == "Recording" \
                    or old_msg["msg_type"] == "Video" \
                    or old_msg["msg_type"] == "Attachment":
                file = '@fil@%s' % (old_msg['msg_content'])
                itchat.send(msg=file, toUserName='filehelper')
                os.remove(old_msg['msg_content'])
            # 删除字典旧消息
            msg_information.pop(old_msg_id)


itchat.auto_login(hotReload=True)
itchat.run()
```