**Python微信库: itchat**

2016年11月08日 14:35:11

<http://www.nulll.me/index.php/archives/72.html?utm_source=tuicool&utm_medium=referral>

开源项目地址：<https://github.com/littlecodersh/ItChat>

在论坛上看到了用Python登录微信并实现自动签到，才了解到一个新的Python库： itchat

库文档说明链接在这： [***itchat***](http://itchat.readthedocs.io/zh/latest/)

我存个档在我网站（主要是我打开很慢），以便以后阅读。

**0x01 Start**

最简单的回复

通过如下代码，可以完成回复所有文本信息（包括群聊）。

1. **import** itchat
2. **from** itchat.content **import** TEXT
4. @itchat.msg\_register
5. **def** **simple\_reply**(msg):
6. **if** msg['Type'] == TEXT:
7. **return** 'I received: %s' % msg['Content']
9. itchat.auto\_login()
10. itchat.run()

常用消息的配置

itchat支持所有的消息类型与群聊，下面的示例中演示了对于这些消息类型简单的配置。

1. #coding=utf8
2. **import** itchat
3. **from** itchat.content **import** \*
5. @itchat.msg\_register([TEXT, MAP, CARD, NOTE, SHARING])
6. **def** **text\_reply**(msg):
7. itchat.send('%s: %s' % (msg['Type'], msg['Text']), msg['FromUserName'])
9. # 以下四类的消息的Text键下存放了用于下载消息内容的方法，传入文件地址即可
10. @itchat.msg\_register([PICTURE, RECORDING, ATTACHMENT, VIDEO])
11. **def** **download\_files**(msg):
12. msg['Text'](msg['FileName'])
13. **return** '@%s@%s' % ({'Picture': 'img', 'Video': 'vid'}.get(msg['Type'], 'fil'), msg['FileName'])
15. # 收到好友邀请自动添加好友
16. @itchat.msg\_register(FRIENDS)
17. **def** **add\_friend**(msg):
18. itchat.add\_friend(\*\*msg['Text']) # 该操作会自动将新好友的消息录入，不需要重载通讯录，微信不要开启“加好友无需验证”
19. itchat.send\_msg('Nice to meet you!', msg['RecommendInfo']['UserName'])
21. # 在注册时增加isGroupChat=True将判定为群聊回复
22. @itchat.msg\_register(TEXT, isGroupChat = True)
23. **def** **groupchat\_reply**(msg):
24. **if** msg['isAt']:
25. itchat.send(u'@%s\u2005I received: %s' % (msg['ActualNickName'], msg['Content']), msg['FromUserName'])
27. itchat.auto\_login(**True**)
28. itchat.run()

当然这里不需要深究为什么这些东西可以这么写，我在这里放出了示例程序只是为了给你一个该sdk相关代码大概样子的概念。

有了大概的模式的了解之后我们就可以进入下一部分的介绍。

**0x02 Login**

在上一部分中你看到了基本的注册与登陆，而显然登陆使用的是itchat提供了auto\_login方法，调用即可完成登录。

一般而言，我们都会在完成消息的注册后登陆。

当然这里需要特别强调的是三点，分别是短时间关闭重连、命令行二维码与自定义登陆内容。 itchat提供了登陆状态暂存，关闭程序后一定时间内不需要扫码即可登录。 为了方便在无图形界面使用itchat，程序内置了命令行二维码的显示。 \* 如果你需要就登录状态就一些修改（例如更改提示语、二维码出现后邮件发送等）。

\*\*0x01-1 短时间关闭程序后重连\*\*

这样即使程序关闭，一定时间内重新开启也可以不用重新扫码。

最简单的用法就是给 **auto\_login** 方法传入值为真的 **hotReload** 。

该方法会生成一个静态文件 **itchat.pkl** ，用于存储登陆的状态。

1. **import** itchat
2. **from** itchat.content **import** TEXT
4. @itchat.msg\_register(TEXT)
5. **def** **simple\_reply**(msg):
6. print(msg['Text'])
8. itchat.auto\_login(hotReload=**True**)
9. itchat.run()
10. itchat.dump\_login\_status()

通过设置statusStorageDir可以将静态文件指定为其他的值。

这一内置选项其实就相当于使用了以下两个函数的这一段程序：

1. **import** itchat
2. **from** itchat.content **import** TEXT
4. **if** itchat.load\_login\_status():
5. @itchat.msg\_register(TEXT)
6. **def** **simple\_reply**(msg):
7. print(msg['Text'])
8. itchat.run()
9. itchat.dump\_login\_status()
10. **else**:
11. itchat.auto\_login()
12. itchat.dump\_login\_status()
13. print('Config stored, so exit.')

其中load\_login\_status与dump\_login\_status分别对应读取与导出设置。

通过设置传入的fileDir的值可以设定导入导出的文件。

\*\*0x01-2 命令行二维码显示\*\*

通过以下命令可以在登陆的时候使用命令行显示二维码：

**itchat**.auto\_login(enableCmdQR=True)

部分系统可能字幅宽度有出入，可以通过将enableCmdQR赋值为特定的倍数进行调整：

# 如部分的linux系统，块字符的宽度为一个字符（正常应为两字符），故赋值为2

itchat.auto\_login(enableCmdQR=2)

默认控制台背景色为暗色（黑色），若背景色为浅色（白色），可以将enableCmdQR赋值为负值：

**itchat**.auto\_login(enableCmdQR=-1)

\*\*0x01-2 自定义登录过程\*\*

如果需要控制登录的过程，可以阅读下面的内容。

同时itchat也提供了登陆所需的每一步的方法，登陆的过程按顺序为： 获取二维码uuid->获取二维码->判断是否已经登陆成功->获取初始化数据->更新微信相关信息（通讯录、手机登陆状态）->循环扫描新信息（开启心跳）

获取二维码uuid

获取生成二维码所需的uuid，并返回。

* 方法名称： **get\_QRuuid**
* 所需值：无
* 返回值：成功->uuid，失败->None

获取二维码

根据uuid获取二维码并打开，返回是否成功。

* 方法名称： **get\_QR**
* 所需值：uuid
* 返回值：成功->True，失败->False

判断是否已经登陆成功

判断是否已经登陆成功，返回扫描的状态码。

* 方法名称： **check\_login**
* 所需值：uuid
* 返回值：登陆成功->'200'，已扫描二维码->'201'，二维码失效->'408'，未获取到信息->'0'

获取初始化数据

获取微信用户信息以及心跳所需要的数据。

* 方法名称： **web\_init**
* 所需值：无
* 返回值：存储登录微信用户信息的字典

获取微信通讯录

获取微信的所有好友信息并更新。

* 方法名称： **get\_contract**
* 所需值：无
* 返回值：存储好友信息的列表

更新微信手机登陆状态

在手机上显示登录状态。

* 方法名称： **show\_mobile\_login**
* 所需值：无
* 返回值：无

循环扫描新信息（开启心跳）

循环扫描是否有新的消息，开启心跳包。

* 方法名称： **start\_receiving**
* 所需值：无
* 返回值：无

**EG：**

一个登录例子：

1. **import** itchat, time, sys
3. **def** **output\_info**(msg):
4. print('[INFO] %s' % msg)
6. **def** **open\_QR**():
7. **for** get\_count **in** range(10):
8. output\_info('Getting uuid')
9. uuid = itchat.get\_QRuuid()
10. **while** uuid **is** **None**: uuid = itchat.get\_QRuuid();time.sleep(1)
11. output\_info('Getting QR Code')
12. **if** itchat.get\_QR(uuid): **break**
13. **elif** get\_count >= 9:
14. output\_info('Failed to get QR Code, please restart the program')
15. sys.exit()
16. output\_info('Please scan the QR Code')
17. **return** uuid
19. uuid = open\_QR()
20. waitForConfirm = **False**
21. **while** 1:
22. status = itchat.check\_login(uuid)
23. **if** status == '200':
24. **break**
25. **elif** status == '201':
26. **if** waitForConfirm:
27. output\_info('Please press confirm')
28. waitForConfirm = **True**
29. **elif** status == '408':
30. output\_info('Reloading QR Code')
31. uuid = open\_QR()
32. waitForConfirm = **False**
33. userInfo = itchat.web\_init()
34. itchat.show\_mobile\_login()
35. itchat.get\_contract()
36. output\_info('Login successfully as %s'%userInfo['NickName'])
37. itchat.start\_receiving()
39. # Start auto-replying
40. @itchat.msg\_register
41. **def** **simple\_reply**(msg):
42. **if** msg['Type'] == 'Text':
43. **return** 'I received: %s' % msg['Content']
44. itchat.run()

**0x03 Register**

**注册消息方法**

itchat将根据接收到的消息类型寻找对应的已经注册的方法。

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。

在运行过程当中也可以动态注册方法，注册方式与结果不变。

**注册**

你可以通过两种方式注册消息方法

1. **import** itchat
2. **from** itchat.content **import** \*
4. # 不带参数注册，所有消息类型都将调用该方法（包括群消息）
5. @itchat.msg\_register
6. **def** **simple\_reply**(msg):
7. **if** msg['Type'] == 'Text':
8. **return** 'I received: %s' % msg['Text']
10. # 带参数注册，该类消息类型将调用该方法
11. @itchat.msg\_register([TEXT, MAP, CARD, NOTE, SHARING])
12. **def** **text\_reply**(msg):
13. itchat.send('%s: %s' % (msg['Type'], msg['Text']), msg['FromUserName'])

**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。

本api增加Text、Type（也就是参数）键值，方便操作。

itchat.content中包含所有的消息类型参数，内容如下表所示：

比如你需要存储发送给你的附件：

1. @itchat.msg\_register(ATTACHMENT)
2. **def** **download\_files**(msg):
3. msg['Text'](msg['FileName'])

值得注意的是，群消息增加了三个键值： isAt: 判断是否@本号 ActualNickName: 实际NickName \* Content: 实际Content

可以通过本程序测试：

1. **import** itchat
2. **from** itchat.content **import** TEXT
4. @itchat.msg\_register(TEXT, isGroupChat = True)
5. **def** **text\_reply**(msg):
6. print(msg['isAt'])
7. print(msg['ActualNickName'])
8. print(msg['Content'])
10. itchat.auto\_login()
11. itchat.run()

**注册消息的优先级**

优先级分别为：后注册消息先于先注册消息，带参数消息先于不带参数消息。

以下面的两个程序为例：

1. **import** itchat
2. **from** itchat.content **import** \*
4. itchat.auto\_login()
6. @itchat.msg\_register(TEXT)
7. **def** **text\_reply**(msg):
8. **return** 'This is the old register'
10. @itchat.msg\_register(TEXT)
11. **def** **text\_reply**(msg):
12. **return** 'This is a new one'
14. itchat.run()

在私聊发送文本时将会回复This is a new one。

1. **import** itchat
2. **from** itchat.content **import** \*
4. itchat.auto\_login()
6. @itchat.msg\_register
7. **def** **general\_reply**(msg):
8. **return** 'I received a %s' % msg['Type']
10. @itchat.msg\_register(TEXT)
11. **def** **text\_reply**(msg):
12. **return** 'You said to me one to one: %s' % msg['Text']
14. itchat.run()

仅在私聊发送文本时将会回复You said to me one to one，其余情况将会回复I received a ...。

**动态注册消息**

动态注册时可以选择将 **itchat.run()** 放入另一线程或使用 **configured\_reply()** 方法处理消息。

两种方法分别是：

1. # 使用另一线程，但注意不要让程序运行终止
2. **import** thread
4. thread.start\_new\_thread(itchat.run, ())
6. # 使用configured\_reply方法
7. **while** 1:
8. itchat.configured\_reply()
9. # some other functions
10. time.sleep(1)

以下给出一个动态注册的例子：

1. #coding=utf8
2. **import** thread
4. **import** itchat
5. **from** itchat.content **import** \*
7. replyToGroupChat = **True**
8. functionStatus = **False**
10. **def** **change\_function**():
11. **if** replyToGroupChat != functionStatus:
12. **if** replyToGroupChat:
13. @itchat.msg\_register(TEXT, isGroupChat = True)
14. **def** **group\_text\_reply**(msg):
15. **if** u'关闭' **in** msg['Text']:
16. replyToGroupChat = **False**
17. **return** u'已关闭'
18. **elif** u'开启' **in** msg['Text']:
19. **return** u'已经在运行'
20. **return** u'输入"关闭"或者"开启"测试功能'
21. **else**:
22. @itchat.msg\_register(TEXT, isGroupChat = True)
23. **def** **group\_text\_reply**(msg):
24. **if** u'开启' **in** msg['Text']:
25. replyToGroupChat = **True**
26. **return** u'重新开启成功'
27. functionStatus = replyToGroupChat
29. thread.start\_new\_thread(itchat.run, ())
31. **while** 1:
32. change\_function()
33. time.sleep(.1)

**0x04 Reply**

**回复**

itchat提供五种回复方法，建议直接使用send方法。

**send方法**

* 方法：

send(msg='Text Message', toUserName=None)

* 所需值：
* 1.msg：消息内容
* 2.'@fil@文件地址'将会被识别为传送文件，'@img@图片地址'将会被识别为传送图片，'@vid@视频地址'将会被识别为小视频

3.toUserName：发送对象，如果留空将会发送给自己

* 返回值：发送成功->True, 失败->False

#coding=utf8

**import** itchat

itchat.auto\_login()

itchat.send('Hello world!')

# 请确保该程序目录下存在：gz.gif以及xlsx.xlsx

itchat.send('@img@%s' % 'gz.gif')

itchat.send('@fil@%s' % 'xlsx.xlsx')

itchat.send('@vid@%s' % 'demo.mp4')

**send\_msg方法**

* 方法：

send\_msg(msg='Text Message', toUserName=None)

* 所需值：
* msg：消息内容

toUserName：发送对象，如果留空将会发送给自己

* 返回值：发送成功->True, 失败->False
* 程序示例：

import itchat

itchat.auto\_login()

itchat.send\_msg('Hello world')

**send\_file方法**

* 方法：

send\_file(fileDir, toUserName=None)

* 所需值：
* fileDir：文件路径（不存在该文件时将打印无此文件的提醒）

toUserName：发送对象，如果留空将会发送给自己

* 返回值：发送成功->True, 失败->False

#coding=utf8

import itchat

itchat.auto\_login()

#请确保该程序目录下存在：xlsx.xlsx

itchat.send\_file('xlsx.xlsx')

**send\_img方法**

* 方法：

send\_img(fileDir, toUserName=None)

* 所需值：
* fileDir：文件路径（不存在该文件时将打印无此文件的提醒）

toUserName：发送对象，如果留空将会发送给自己

* 返回值：发送成功->True, 失败->False

#coding=utf8

**import** itchat

itchat.auto\_login()

# 请确保该程序目录下存在：gz.gif

itchat.send\_img('gz.gif')

**send\_video方法**

* 方法：

send\_video(fileDir, toUserName=None)

* 所需值：
* fileDir：文件路径（不存在该文件时将打印无此文件的提醒）

toUserName：发送对象，如果留空将会发送给自己

* 返回值：发送成功->True, 失败->False

需要保证发送的视频为一个实质的mp4文件

#coding=utf8

import itchat

itchat.auto\_login()

#请确保该程序目录下存在：demo.mp4

itchat.send\_file('demo.mp4')

**0x05 Memmber stuff**

在使用个人微信的过程当中主要有三种账号需要获取，分别为： 好友 公众号 \* 群聊

itchat为这三种账号都提供了整体获取方法与搜索方法。

而群聊多出获取用户列表方法以及创建群聊、增加、删除用户的方法。

这里我们分这三种分别介绍如何使用。

**好友**

好友的获取方法为 **get\_friends** ，将会返回完整的好友列表。 其中每个好友为一个字典 列表的第一项为本人的账号信息 \* 传入update键为True将可以更新好友列表并返回

好友的搜索方法为 **search\_friends** ，有四种搜索方式： 1. 仅获取自己的用户信息 2. 获取特定 **UserName** 的用户信息 3. 获取备注、微信号、昵称中的任何一项等于 **name** 键值的用户 4. 获取备注、微信号、昵称分别等于相应键值的用户

其中三、四项可以一同使用，下面是示例程序：

# 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

**itchat**.search\_friends(userName='@abcdefg1234567')

# 获取任何一项等于**name**键值的用户

**itchat**.search\_friends(name='littlecodersh')

# 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')

**公众号**

公众号的获取方法为 **get\_mps** ，将会返回完整的公众号列表。 其中每个公众号为一个字典 传入update键为True将可以更新公众号列表并返回

公众号的搜索方法为 **search\_mps** ，有两种搜索方法： 1. 获取特定 **UserName** 的公众号 2. 获取名字中含有特定字符的公众号

如果两项都做了特定，将会仅返回特定 **UserName** 的公众号，下面是示例程序：

# 获取特定**UserName**的公众号，返回值为一个字典

**itchat**.search\_mps(userName='@abcdefg1234567')

# 获取名字中含有特定字符的公众号，返回值为一个字典的列表

**itcaht**.search\_mps(name='LittleCoder')

# 以下方法相当于仅特定了**UserName**

**itchat**.search\_mps(userName='@abcdefg1234567', name='LittleCoder')

**群聊**

群聊的获取方法为 **get\_chatrooms** ，将会返回完整的群聊列表。 其中每个群聊为一个字典 传入update键为True将可以更新群聊列表并返回

群聊的搜索方法为 **search\_chatrooms** ，有两种搜索方法： 1. 获取特定UserName的群聊 2. 获取名字中含有特定字符的群聊

如果两项都做了特定，将会仅返回特定UserName的群聊，下面是示例程序：

# 获取特定**UserName**的群聊，返回值为一个字典

**itchat**.search\_chatrooms(userName='@abcdefg1234567')

# 获取名字中含有特定字符的群聊，返回值为一个字典的列表

**itcaht**.search\_chatrooms(name='LittleCoder')

# 以下方法相当于仅特定了**UserName**

**itchat**.search\_chatrooms(userName='@abcdefg1234567', name='LittleCoder')

群聊用户列表的获取方法为 **update\_chatroom** 。 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的成员 该方法需要传入群聊的UserName，返回特定群聊的用户列表

**memberList** = itchat.update\_chatroom('@abcdefg1234567')

创建群聊、增加、删除群聊用户的方法如下所示： 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群管理员，否则会失败

memberList = itchat.get\_friends()[1:]

# 创建群聊，topic键值为群聊名

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])

**0x06 QAQ**

Q: 为什么我在设定了itchat.auto\_login()的enableCmdQR为True后还是没有办法在命令行显示二维码？

A: 这是由于没有安装可选的包 **pillow** ，可以使用右边的命令安装： **pip install pillow**

**0x07 Eg**

1. **def** **signin**():
2. # 查找公众号，进行签到
3. user = itchat.search\_mps(name='Nulll.me')
4. UserName = user[0]['UserName']
5. itchat.send(msg=u'3', toUserName=UserName)
6. itchat.dump\_login\_status()
7. pickleDumps('flag', localDay) # 如果执行成功写入标致文件
8. exit()

11. **if** \_\_name\_\_ == '\_\_main\_\_':
12. # 如果不是在登陆状态，就循环登陆
13. **while** **not** itchat.load\_login\_status():
14. sendMail()
15. itchat.auto\_login(hotReload=**True**)
16. itchat.dump\_login\_status()
17. signin() # 签到
18. time.sleep(3600)
19. signin() # 签到

Thank you for read

End!