

wxauto调用文档（适用微信版本v3.9.8.15）

wxauto调用文档（适用微信版本v3.9.8.15）

- 一、版本对比
- 二、获取wxauto
 - 1. 下载项目文件
 - 2. 安装依赖
 - 3. 测试运行
- 三、方法说明
 - 1. 获取当前聊天窗口的聊天记录
 - 2. 打开指定好友聊天窗口
 - 3. 发送消息
 - 4. 发送文件、图片
 - 5. 获取所有未读消息内容
 - 6. 获取一个未读消息内容
 - 7. 获取当前聊天窗口名
 - 8. 加载当前聊天页面更多聊天信息
 - 9. 监听指定好友（群聊）消息
 - 9.1 添加监听对象
 - 9.2 获取监听对象的新消息
 - 10. 接受新的好友申请
 - 10.1 获取新的好友申请对象列表
 - 10.2 通过好友申请对象接受好友请求
 - 11. 切换微信页面
 - 11.1 切换到聊天页面
 - 11.2 切换到通讯录页面
- 四、使用案例
 - 1. 监听指定群或好友消息并回复收到
 - 2. 监听所有未被屏蔽的新消息

更新日期：2024年02月06日

一、版本对比

脚本适用微信3.9.8.15版本，使用前请先检查自己电脑微信是否为该版本，版本不同可能由于UI问题导致某些功能无法正常调用，如果版本不同但是功能可用的话也不必过多纠结版本号的warning信息

二、获取wxauto

1. 下载项目文件

```
git clone https://github.com/cluic/wxauto.git
cd wxauto
```

如clone失败，可在github手动下载zip文件

2. 安装依赖

```
pip install -r requirements.txt
```

等待安装完成即可

3. 测试运行

打开cmd，运行demo.py：

```
python demo.py
```

如果自动发送并打印了当前页面的聊天记录出来，并且最后输出“wxauto测试完成！”，则测试完成，可以继续调用wxauto来完成您的项目

三、方法说明

假设您已经完成了上面的测试，可以正常运行wxauto脚本

```
# 导入
>>> from wxauto import WeChat

# 获取微信窗口对象
>>> wx = WeChat()
初始化成功，获取到已登录窗口：xxxx
```

上面定义了wx变量，下述文档不再重复定义和解释wx变量

1. 获取当前聊天窗口的聊天记录

```
# 获取当前窗口聊天记录，并自动保存聊天图片
>>> msgs = wx.GetAllMessage(savepic=True)
```

wx.GetAllMessage

方法说明：

获取当前窗口中加载的所有聊天记录

参数：

参数名	类型	默认值	说明
savepic	bool	False	是否自动保存聊天图片

2. 打开指定好友聊天窗口

```
# 打开“文件传输助手”聊天窗口
>>> who = '文件传输助手'
>>> wx.ChatWith(who)
```

wx.ChatWith

方法说明：

打开指定好友（群组）聊天窗口

参数：

参数名	类型	默认值	说明
who	str	/	要打开的聊天框好友名或群名

3. 发送消息

```
# 给“文件传输助手”发送消息
>>> who = '文件传输助手'
>>> msg = '''这是一条消息
这是第二行
这是第三行
'''
>>> wx.SendMsg(msg, who=who)
```

wx.SendMsg

方法说明：

给指定人员（群组）发送消息

参数：

参数名	类型	默认值	说明
msg	str	/	要发送的文字内容
who	str	None	要发送给谁，默认则发送给当前打开的页面
clear	bool	True	是否清除原本聊天编辑框的内容

4. 发送文件、图片

```
# 给“文件传输助手”发送文件（图片同理）
>>> who = '文件传输助手'
# 指定文件路径（绝对路径）
>>> files = ['D:/test/test1.txt', 'D:/test/test2.txt', 'D:/test/test3.txt']
>>> wx.SendFiles(self, files, who=who)
```

wx.SendFiles

方法说明：

给指定人员（群组）发送文件或者图片

参数：

参数名	类型	默认值	说明
filepath	str list	/	指定文件路径，单个文件str，多个文件list
who	str	None	要发送给谁，默认则发送给当前打开的页面

5. 获取所有未读消息内容

```
>>> msgs = wx.GetAllNewMessage()
```

wx.GetAllNewMessage

方法说明：

获取所有未读消息的内容，即存在未读数量小圆点的聊天窗

注：该方法暂时只能读取未开启消息免打扰的好友的未读消息，开启消息免打扰的聊天无法获取

6. 获取一个未读消息内容

```
>>> msgs = wx.GetNextNewMessage()
>>> msgs
{'张三': [['张三', '哈哈', '42373591784181']]}
```

wx.GetNextNewMessage

方法说明：

只获取一个未读消息内容，这样多个聊天对象有新消息时，可以逐一获取消息内容并进行回复

参数名	类型	默认值	说明
savepic	bool	False	是否保存聊天图片

注：该方法暂时只能读取未开启消息免打扰的好友的未读消息，开启消息免打扰的聊天无法获取

7. 获取当前聊天窗口名

```
>>> current = wx.CurrentChat()
```

wx.CurrentChat

方法说明：

获取当前聊天窗口名，即聊天窗口最上方的那个名字

注：该方法获取到的名字，如果是群组，则会带有群组人数，比如：闲置群（352）

8. 加载当前聊天页面更多聊天信息

```
>>> wx.LoadMoreMessage()
```

wx.LoadMoreMessage

方法说明：

利用鼠标滚动加载当前聊天页面更多聊天信息

9. 监听指定好友（群聊）消息

9.1 添加监听对象

```
>>> wx.AddListenChat(who='张三', savepic=True)
```

wx.AddListenChat

方法说明：

将指定聊天对象独立出来，并且加入监听列表中

参数名	类型	默认值	说明
who	str	/	好友名/群名
savepic	bool	False	是否保存聊天图片

9.2 获取监听对象的新消息

```
>>> msgs = wx.GetListenMessage()
>>> msgs
{'张三': [['张三', '哈哈', '42373591784181']], '李四': [['李四', '哈哈', '42373591784256']], '李白': []}
```

10. 接受新的好友申请

10.1 获取新的好友申请对象列表

```
>>> new = wx.GetNewFriends()
>>> new
[<wxauto New Friends Element at 0x1e95fced080 (张三：你好,我是xxx群的张三)>,
<wxauto New Friends Element at 0x1e95fced081 (李四：你好,我是xxx群的李四)>]
```

方法说明：

获取好友申请列表中，状态为可接受的好友申请对象

10.2 通过好友申请对象接受好友请求

```
# 获取第一个可接受的新好友对象
>>> new_friend1 = new[0]
>>> print(new_friend1.name) # 获取好友申请昵称
张三
>>> print(new_friend1.msg) # 获取好友申请信息
你好,我是xxx群的张三

# 接受好友请求, 并且添加备注“备注张三”、添加标签wxauto
>>> new_friend1.Accept(remark='备注张三', tags=['wxauto'])
```



注：该方法接受好友请求后，并不会自动切换回聊天页面，需要配合调用11.1切换至聊天页面，否则其他有关聊天页面的方法不可使用

11. 切换微信页面

11.1 切换到聊天页面

```
>>> wx.SwitchToChat()
```

11.2 切换到通讯录页面

```
>>> wx.SwitchToContact()
```

四、使用案例

1. 监听指定群或好友消息并回复收到

```
from wxauto import weChat
import time

# 实例化微信对象
wx = weChat()

# 指定监听目标
listen_list = [
    '张三',
    '李四',
    '工作群A',
```

```

    '工作群B'
]
for i in listen_list:
    wx.AddListenChat(who=i, savepic=True) # 添加监听对象并且自动保存新消息图片

# 持续监听消息，并且收到消息后回复“收到”
wait = 10 # 设置10秒查看一次是否有新消息
while True:
    msgs = GetListenMessage()
    for i in msgs:
        # =====
        # 处理消息逻辑
        #
        # 处理消息内容的逻辑每个人都不同，按自己想法写就好了，这里不写了
        #
        # =====

        # 如果获取到新消息了，则回复收到
        if msgs[i]:
            wx.SendMsg(msg='收到', who=i) # 回复收到

```

2. 监听所有未被屏蔽的新消息

```

from wxauto import WeChat
import time

# 实例化微信对象
wx = WeChat()

# 持续监听消息，并且收到消息后回复“收到”
wait = 10 # 设置10秒查看一次是否有新消息
while True:
    msg = wx.GetNextNewMessage(savepic=True)
    # 如果获取到新消息了，则回复收到
    if msg:
        # =====
        # 处理消息逻辑
        #
        # 处理消息内容的逻辑每个人都不同，按自己想法写就好了，这里不写了
        #
        # =====

        wx.SendMsg(msg='收到', who=list(msg)[0]) # 回复收到

```