

wxautoapi文档

wxautoapi文档

开始使用

快速开始:

1. 调用实例
2. 多开实例

一、个人模块

1. 查找人
 - (1) self.find_persons
 - (2) self.find_person
2. 方法
 - (1) 发送文本消息 (send_text)
 - (2) 发送图片 (send_image)
 - (3) 发送文件 (send_file)
 - (4) 发送 XML 卡片 (send_xml_card)
 - (5) 置顶 (sticky_top)
 - (6) 取消置顶 (remove_top)
 - (7) 设置备注 (set_remark)

二、群组模块

1. 查找群组
2. 方法
 - (1-7) 同个人
 - (8) 获取群组成员 (get_members)
 - (9) 修改群组名称 (modify_name)
 - (10) *发送群组通知 (send_notice)
 - (11) 添加群组成员 (add_member)
 - (12) 删除群组成员 (remove_member)
 - (13) 获取群组人数 (get_number)
 - (14) 保存群组至通讯录 (save_contact)
 - (15) 删除通讯录群组 (remove_contact)
 - (16) @群成员 (at)
 - (17) @所有人 (at_all)
 - (18) 退出群组 (quit)

三、消息模块

1. 接收消息 (getmsgs)
2. 消息对象汇总
 - ① 文本消息 (TextMessage)
 - ② 图片消息 (ImageMessage)
 - ③ 文件消息 (FileMessage)
 - ④ 好友请求消息 (NewFriendMessage)
 - ⑤ 撤回消息 (RecallMessage)
 - ⑥ 其他消息 (OtherMessage)

声明

微信版本: 3.7.0.29

更新日期: 2022.07.08

版本: 测试版, 请勿转卖

开始使用

注意：该项目只支持Windows PC微信3.7.0.29版本，调用前请先检查版本！！

注意：该项目只支持Windows PC微信3.7.0.29版本，调用前请先检查版本！！

注意：该项目只支持Windows PC微信3.7.0.29版本，调用前请先检查版本！！



快速开始：

1. 调用实例

```
>>> from wxautoapi import WeChat

# 登录微信（已登录的微信请先退出）
>>> wx = WeChat()
成功启动微信：1804，等待登录...
微信已登录 (<wxid_xxxxxxxxxxxx>, XXXX)>>>>>>

# ===== 注意！ =====

# 请确保微信已登录后，再运行下述步骤，否则无法正常调用相关方法！！
# 请确保微信已登录后，再运行下述步骤，否则无法正常调用相关方法！！
# 请确保微信已登录后，再运行下述步骤，否则无法正常调用相关方法！！

# ===== 注意！ =====

# 文件传输助手
>>> wx.filehelper.send_text('你好！')
>>> wx.filehelper.send_file('D:/test.txt')

# 查找好友
# 方法一（不知道wxid）：查找备注为张三的好友
>>> person = wx.find_person(by='remark', value='张三')
>>> person.send_text('你好~')
# 方法二（知道wxid）：
>>> person = wx['wxid_xxxxxxxx']

# 查找群（查找备注为工作1群的微信群）
>>> group = wx.find_group(by='remark', value='工作1群')
>>> group.send_text('大家好')

# 获取消息
>>> Messages = wx.getmsgs()
```

```
>>> Messages # 返回一个消息列表，包含所有消息对象
[<WeChat TextMessage Element at 0x2a2781e16a0>,
 <WeChat ImageMessage Element at 0x2a276d7dc18>,
 <WeChat Other Message Element at 0x2a276d7dba8>,
 <WeChat TextMessage Element at 0x2a276d7dc50>,
 <WeChat TextMessage Element at 0x2a276d7d908>]
>>> msg = Messages[0]
>>> msg.text # 消息内容
"xxxxx"
```

2. 多开实例

```
>>> from wxautoapi import WeChat

>>> n = 3 # 多开数量，例如同时开启3个微信
>>> wxdict = {f'wx{i}':WeChat() for i in range(n)} #
>>> wxdict
{'wx0': <WeChat Object at 0x2a275e15f28 - (未登录)>,
 'wx1': <WeChat Object at 0x2a276804320 - (未登录)>,
 'wx2': <WeChat Object at 0x2a27681cbe0 - (未登录)>}
# 登录微信之后，所有操作与调用实例一样
>>> wxdict['wx0'].filehelper.send_text('123') # 操作第一个微信向文件传输助手发送消息：
123
```

```
In [1]: >>> from wxautoapi import WeChat
...: M
...: n = 3 # 多开数量，例如同时开启3个微信
...: # 多开
...: wxdict = {f'wx{i}':WeChat() for i in range(n)}
== 测试版本V1.0.1 ==
Github: https://github.com/cluic/wxauto

>>> 成功启动微信：55556，等待登录...
>>> 成功启动微信：32840，等待登录...
>>> 成功启动微信：49336，等待登录...

In [2]:
```



一、个人模块

该模块为个人模块，仅 `wx.find_person` 方法返回的群组元素调用

```
>>> person = wx.find_person('nickname', '张三')
>>> person
<WeChat Person Element at 0x23ff6ece400: (wxid_xxxxxxxxx - 张三)>
```

1. 查找人

(1) self.find_persons

```
def find_persons(self, by: str, value: str) -> list:
    ...
```

参数说明:

- by
用哪种方法查找，可选择 remark（备注），nickname（微信昵称），wxid（微信号）。
- value
by的值，例如：by为remark，value即要查找的微信好友的备注名

返回值:

列表：所有符合条件的微信好友列表

(2) self.find_person

```
def find_person(self, by: str, value: str) -> Person:
    ...
```

参数说明:

同 self.find_persons

返回值:

Person：如果存在符合条件的微信好友，默认返回第一个好友

2. 方法

(1) 发送文本消息 (send_text)

```
def send_text(self, msg: str):
    ...
```

参数说明:

- msg
要发送的文本内容

示例代码:

```
# 向张三发送文本消息：你好~
>>> person.send_text('你好~')
```

(2) 发送图片 (send_image)

```
def send_image(self, img: str):  
    ...
```

参数说明:

- img
要发送的图片绝对地址, 例如: "D:/image/test.png"

示例代码:

```
# 向张三发送图片:  
>>> person.send_image("D:/image/test.png")
```

(3) 发送文件 (send_file)

```
def send_file(self, file: str):  
    ...
```

参数说明:

- file
要发送的文件绝对地址, 例如: "D:/files/test.rar"

示例代码:

```
# 向张三发送文件:  
>>> person.send_file("D:/files/test.rar")
```

(4) 发送 XML 卡片 (send_xml_card)

```
def send_xml_card(self, title: str, desc: str, link: str, imagepath: str):  
    ...
```

参数说明:

- title
卡片标题
- desc
卡片描述
- link
卡片链接地址
- imagepath
卡片展示的图片绝对地址 (图片不易过大, 具体限制不太清楚)

示例代码:

```
# 向张三发送一个自定义XML卡片:
>>> person.send_xml_card(
...     title = "这是标题",
...     desc = "这里是卡片描述",
...     link = "http://baidu.com",
...     imagepath = "D:/image/test.png"
... )
```

(5) 置顶 (sticky_top)

示例代码:

```
>>> person.sticky_top()
```

(6) 取消置顶 (remove_top)

示例代码:

```
>>> person.remove_top()
```

(7) 设置备注 (set_remark)

```
def set_remark(self, remark: str):
    ...
```

参数说明:

- remark
要设置的备注名

示例代码:

```
>>> person.set_remark('皮卡丘')
```

二、群组模块

该模块为群组模块, 仅wx.find_group方法返回的群组元素调用

```
>>> group = wx.find_group('nickname', '工作群')
>>> group
<wechat Group Element at 0x23ff854fd30: (xxxxxxxx@chatroom - 工作群)>
```

1. 查找群组

与个人方法相同

2. 方法

(1-7) 同个人

与个人 (1) - (7) 方法相同

(8) 获取群组成员 (get_members)

示例代码:

```
>>> group.get_members()
```

(9) 修改群组名称 (modify_name)

```
def modify_name(self, name:str):  
    ...
```

参数说明:

- name
准备修改的群名

示例代码:

```
# 将群名修改为: xx交流群  
>>> group.modify_name("xx交流群")
```

(10) *发送群组通知 (send_notice)

```
def send_notice(self, msg: str):  
    ...
```

参数说明:

- msg
要发送的通知内容

示例代码:

```
>>> group.send_notice("通知: xxxxxx")
```

*注: 该方法不太稳定, 不建议使用

(11) 添加群组成员 (add_member)

```
def add_member(self, wxids: list):  
    ...
```

参数说明:

- wxids

群组成员的微信id列表

示例代码：

```
>>> group = wx.find_group('remark', '工作群')    # 指定一个群组
>>> person1 = wx.find_person('remark', '张三')   # 选择要拉的人1
>>> person2 = wx.find_person('remark', '李四')   # 选择要拉的人2
>>> group.add_member([person1, person2])         # 拉入群组
```

(12) 删除群组成员 (remove_member)

```
def remove_member(self, wxids: list):
    ...
```

参数说明：

- wxids
群组成员的微信id列表

(13) 获取群组人数 (get_number)

```
def get_number(self):
    ...
```

(14) 保存群组至通讯录 (save_contact)

```
def save_contact(self):
    ...
```

(15) 删除通讯录群组 (remove_contact)

```
def remove_contact(self):
    ...
```

(16) @群成员 (at)

```
def at(self, members, msg=''):
    ...
```

参数说明：

- members
群组成員列表，列表元素为
- msg
要发送的消息

(17) @所有人 (at_all)

```
def at_all(self, msg):
    ...
```


参数说明:

- msg
要发送的消息

(18) 退出群组 (quit)

```
def quit(self):  
    ...
```

三、消息模块

1. 接收消息 (getmsgs)

```
def getmsgs(self, t=60):  
    ...
```

参数说明:

- t
获取 t 秒内的新消息

返回值:

消息列表, 列表元素为消息对象

示例代码:

```
# 获取消息  
>>> Messages = wx.getmsgs()  
>>> Messages # 返回一个消息列表, 包含所有消息对象  
[<weChat TextMessage Element at 0x2a2781e16a0>,  
 <weChat ImageMessage Element at 0x2a276d7dc18>,  
 <weChat Other Message Element at 0x2a276d7dba8>,  
 <weChat TextMessage Element at 0x2a276d7dc50>,  
 <weChat TextMessage Element at 0x2a276d7d908>]  
>>> msg = Messages[0]  
>>> msg.text # 消息内容  
"xxxxxx"
```

2. 消息对象汇总

① 文本消息 (TextMessage)

```
# 消息类型
>>> msg.msgtype
'text'

# 快速回复文本消息
>>> msg.reply('收到! ')

# 获取消息发送人
>>> msg.sender
<WeChat Person Element at 0x23ff6ece400: (wxid_xxxxxxx - 张三)>
```

② 图片消息 (ImageMessage)

```
# 消息类型
>>> msg.msgtype
'image'

# 快速回复文本消息
>>> msg.reply('收到! ')

# 获取消息发送人
>>> msg.sender
<WeChat Person Element at 0x23ff6ece400: (wxid_xxxxxxx - 张三)>

# 保存图片
>>> msg.save('D:/Image/test.png')
```

③ 文件消息 (FileMessage)

```
# 消息类型
>>> msg.msgtype
'image'

# 快速回复文本消息
>>> msg.reply('收到! ')

# 获取消息发送人
>>> msg.sender
<WeChat Person Element at 0x23ff6ece400: (wxid_xxxxxxx - 张三)>

# 获取文件路径
>>> msg.filepath
"D:/WeChatFiles/Files/test.txt"

# 移动文件到其他文件夹
>>> msg.move('D:/File')

# 打开文件所在的文件夹
>>> msg.open_folder()

# 删除文件
>>> msg.delete()
```

④ 好友请求消息 (NewFriendMessage)

```
# 消息类型
>>> msg.msgtype
'newfriend'

# 接受好友请求
>>> msg.accept()
```

⑤ 撤回消息 (RecallMessage)

```
# 消息类型
>>> msg.msgtype
'recall'

# 撤回的消息内容
>>> msg.content

# 快速回复文本消息
>>> msg.reply('你撤回了一条消息，我已经收到了！')
```

⑥ 其他消息 (OtherMessage)

```
# 消息类型
>>> msg.msgtype
'other'

# 快速回复文本消息
>>> msg.reply('收到！')
```

声明

1. 本项目仅供学习研究，不可用于违法犯罪活动
2. 本项目完全独立，跟作者无任何信息交互，不存在任何盗取数据的行为，如若发现可报警处理