
wxpy Documentation

0.2.1

Youfou

3 12, 2017

Contents

1		3
2		5
3		7
4		9
5		11
5.1	11
5.2	13
5.3	17
5.4	21
5.5	23
5.6	itchat	24
6		25
Python		27

/ API Python

CHAPTER 1

-
-
- -
 -
 -
 -
 -
 -
 - ...¹

¹ ...

CHAPTER 2

Python 3.x

```
pip3 install -U wxpy
```


CHAPTER 3

:

```
#  
from wxpy import *  
#  
bot = Bot()
```

:

```
# ""  
my_friend = bot.friends().search('', sex=MALE, city="")[0]
```

:

```
#  
my_friend.send('Hello WeChat!')  
#  
my_friend.send_image('my_picture.jpg')
```

:

```
#  
@bot.register()  
def print_others(msg):  
    print(msg)  
  
# `my_friend` (!)  
@bot.register(my_friend)  
def reply_my_friend(msg):  
    return 'received: {} ({}).format(msg.text, msg.type)  
  
#  
bot.start()
```


CHAPTER 4

-
-
-
-
- itchat
- -
 -
 - /
 -

CHAPTER 5

Bot Web

:

/

: *Bot*

```
class wxpy.Bot (cache_path=None, console_qr=False, qr_path=None, qr_callback=None, login_callback=None, logout_callback=None)
    Web
```

- **cache_path** –

- **console_qr** –

pillow (*pip3 install pillow*)

(int) 2 True 2

: Linux True 2 macOS Terminal -2

- `qr_path` –
- `qr_callback` – : uuid, status, qrcode
- `login_callback` –
- `logout_callback` –

Bot.`chats` (`update=False`)
`update` –

Bot.`friends` (`update=False`)
`update` –

Bot.`groups` (`update=False, contact_only=False`)

- `update` –
- `contact_only` –

Bot.`mps` (`update=False`)
`update` –

Bot.`add_friend` (`user, verify_content=''`)

- `user` – user_name
- `verify_content` –

Bot.`accept_friend` (`user, verify_content=''`)

- `user` – user_name
- `verify_content` –

::

```
#
@bot.register(FRIENDS)
# 'wxpy'
def auto_accept_friends(msg):
#
    if 'wxpy' in msg.text.lower():
        # (msg.card)
        new_friend = msg.card.accept()
```



```
#
new_friend.send('')
```

Bot.**create_group**(users, topic=None)

- **users** –
- **topic** –

Bot.**user_details**(user_or_users, chunk_size=50)
()

- **user_or_users** – user_name
- **chunk_size** – 50

Bot.**logout**()

Bot chats(), friends() groups(), mps(),

Chats Groups Chats.search():

```
from wxpy import *
bot = Bot()
my_friend = bot.friends().search('', sex=MALE, city='') [0]
# <Friend: >
```

() *send*:

```
#
my_friend.send('Hello, WeChat!')
#
my_friend.send_image('my_picture.png')
#
my_friend.send_video('my_video.mov')
#
my_friend.send_file('my_file.zip')
#
my_friend.send('@img@my_picture.png')
```

```
class wxpy.Chat (raw, bot)
    (User)(Group)
```

```
    bot
```

```
    raw
```

```
    nick_name
```

```
    ()
```

```
    name
```

```
    : ()
```

```
    wxid
```

```
    ()
```

```
    user_name
```

```
    ID
```

```
    : ID
```

```
send (msg, media_id=None)
```

```
    msg
```

```
    • msg –
```

```
        : '@fil@', '@img@', '@msg@', '@vid@' ()
```

```
        :
```

```
        :
```

```
    • media_id –
```

```
send_image (path, media_id=None)
```

```
    • path –
```

```
    • media_id –
```

```
send_file (path, media_id=None)
```

```
    • path –
```

```
    • media_id –
```

```
send_video (path=None, media_id=None)
```

```
    • path –
```

- `media_id` –

```
send_msg (msg='Hello WeChat! – by wxpy')
    msg –
```

```
send_raw_msg (msg_type, content)
    :
```

```
from wxpy import *
bot = Bot()
@bot.register(msg_types=CARD)
def reply_text(msg):
    msg.sender.send_raw_msg(msg['MsgType'], msg['Content'])
```

```
pin ()
```

```
unpin ()
```

```
class wxpy.User (raw, bot)
    (Friend)(Member)(MP)

    remark_name
    sex
    province
    city
    signature
    add (verify_content='')
        verify_content – ()
    accept (verify_content='')
        verify_content – ()

    is_friend
        True False
```

```
class wxpy.Friend (raw, bot)
```

```
class wxpy.Group (raw, bot)
```

members

```
search (name=None, **attributes)
```

- `name` –

- `attributes` –

```
owner
is_owner
    bot
update_group (members_details=False)
    members_details - ()
add_members (users, use_invitation=False)
    • users -
    • use_invitation -
remove_members (members)
    members -
rename_group (name)
    name - (32)
```

```
class wxpy.Member (raw, group)
```

```
display_name
```

```
class wxpy.MP (raw, bot)
```

```
Chats search() stats() stats_text():
```

```
bot.friends().stats_text()
# 100 \n\n: 67 (67.0%) \n: 23 (23.0%) ...
```

```
class wxpy.Chats (chat_list=None, source=None)
```

```
search (name=)
```

- **name** - ()
- **attributes** - sex(), province(), city() province=""

```
stats (attribs=('sex', 'province', 'city'))
    attribs -
```

```
stats_text (total=True, sex=True, top_provinces=10, top_cities=10)
```

- **total** -

- `sex` -
- `top_provinces` -
- `top_cities` -

`add_all` (*interval=3, verify_content=''*)

- `interval` - ()
- `verify_content` -

`class wxpy.Groups` (*group_list=None*)

`search` (*name=*

- `name` -
- `users` -
- `attributes` - `owner()`, `is_owner()`, `nick_name()`

1. `Bot.messages`
2. `()`

`class wxpy.Message` (*raw, bot*)

`raw`

`bot`
`text`
`type`
:

```
#
TEXT = 'Text'
#
MAP = 'Map'
#
CARD = 'Card'
#
NOTE = 'Note'
#
SHARING = 'Sharing'
#
PICTURE = 'Picture'
#
```

```
RECORDING = 'Recording'
#
ATTACHMENT = 'Attachment'
#
VIDEO = 'Video'
#
FRIENDS = 'Friends'
#
SYSTEM = 'System'
```

is_at

@ True

card

get_file (*save_path=None*)

save_path – *file_name None*

()

file_name

()

reply (...)

Message.sender.send(...)

reply_image (...)

Message.sender.send_image(...)

reply_file (...)

Message.sender.send_file(...)

reply_video (...)

Message.sender.send_video(...)

reply_msg (...)

Message.sender.send_msg(...)

reply_raw_msg (...)

Message.sender.send_raw_msg(...)

img_height

img_width

play_length

url

URL ()

voice_length

id

ID

location

create_time

sender

receiver

member

: :

Bot.**register** (*senders=None, msg_types=None, except_self=True, run_async=True, enabled=True*)

- **senders** –
- **msg_types** – (SYSTEM)
- **except_self** –
- **run_async** –
- **enabled** –

:

:

1. *chats msg_types*
 2. *chats*
 3. *return <> msg.reply(<>)*
-

:

Bot.**start** (*block=True*)
block – False

- ""
- "" @ TEXT
-

:

```
from wxpy import *
bot = Bot()
my_friend = bot.friends().search('')[0]
boring_group = bot.groups().search('')[0]
```

:

```
@bot.register()
def just_print(msg):
    #
    print(msg)
```

""" @ TEXT:

```
@bot.register([my_friend, Group], TEXT)
def auto_reply(msg):
    # @
    if not (isinstance(msg.sender, Group) and not msg.is_at):
        #
        return ': {} ({}>'.format(msg.text, msg.type)
```

""":

```
@bot.register(boring_group)
def ignore(msg):
    #
    return
```

:

```
bot.start()
```

:

:

```
bot.message_configs
# [<MessageConfig: just_print (Async, Enabled)>,
# <MessageConfig: auto_reply (Async, Enabled)>,
# <MessageConfig: ignore (Async, Enabled)>]
```

:

```
bot.message_configs.disable()
```

just_print :

```
bot.message_configs.enable(just_print)
```

:


```
bot.message_configs.enabled
# [<MessageConfig: just_print (Async, Enabled)>]
```

wxpy

embed()

:

embed()

:

```
from wxpy import *
bot = Bot()
embed() #

>>> bot
# Out[1]: <Bot: >
>>> bot.friends()
# Out[2]: [<Friend: >, <Friend: >, <Friend: >]
```

wxpy.**embed**(*local=None, banner='', shell=None*)

Python

bpythonipython python

- **shell**(*str*) –
‘ipython’bpython’python’
None Python
- **local**(*dict*) – *None*
- **banner**(*str*) –

wxpy

:

wxpy

wxpy -h

- **bot1 bot2 bot3...**
 -
 -
 - : *r1 r2*
- **-c / -cache**
 - *wxpy_*.pkl*
 -
 - *-c*
- **-q / -console_qr**
 -
 -
 - *-q 2*
- **-l / -logging_level (L I)**
 -
 - *INFO*
 - *-l DEBUG*
- **-s / -shell**
 - Python
 - *ipythonbpythonpython*
 - Python
 - *-s bpython*
- **-v / -version**
 - *z*
 - *-v*

bot :

```
wxpy bot
```

2:

```
wxpy bot -q 2
```

r1 r2 :

```
wxpy r1 r2
```

:

```
wxpy r1 r2 -c
```

bpython:

```
wxpy r1 r2 -c -s bpython
```

class wxpy.**Tuling** (*api_key=None*)

wxpy
API KEY
API KEY: <http://www.tuling123.com/>

api_key – API KEY (: <http://www.tuling123.com/>)

do_reply (*msg, to_member=True*)

- **msg** – Message
- **to_member** – @

reply_text (*msg, to_member=True*)

- **msg** – Message
- **to_member** – @

wxpy.**mutual_friends** (**args*)
args – (Bot)(Chats)

wxpy.**ensure_one** (*found*)
ValueError

:

```
my_friend = bot.friends().search('')
# <Friend: >
```

`found` –

ResponseError

`wxpy.dont_raise_response_error(func)`
ResponseError

itchat

wxpy
itchat
wxpy itchat

itchat

wxpy `Bot .core.*` itchat
itchat `search_friends` :

```
from wxpy import *  
bot = Bot()  
found = bot.core.search_friends('')
```

```
: itchat wxpy
```

wxpy itchat `.raw`
:

```
from wxpy import *  
bot = Bot()  
a_friend = bot.friends()[0]  
print(a_friend.raw)
```

CHAPTER 6

<https://github.com/youfou/wxpy>

()

- [wxpy]



W

wxpy, 7

A

accept() (wxpy.User), 15
accept_friend() (wxpy.Bot), 12
add() (wxpy.User), 15
add_all() (wxpy.Chats), 17
add_friend() (wxpy.Bot), 12
add_members() (wxpy.Group), 16

B

bot (wxpy.Chat), 14
bot (wxpy.Message), 17

C

card (wxpy.Message), 18
chats() (wxpy.Bot), 12
city (wxpy.User), 15
create_group() (wxpy.Bot), 13
create_time (wxpy.Message), 18

D

display_name (wxpy.Member), 16
do_reply() (wxpy.Tuling), 23

F

file_name (wxpy.Message), 18
friends() (wxpy.Bot), 12

G

get_file() (wxpy.Message), 18
groups() (wxpy.Bot), 12

I

id (wxpy.Message), 18
img_height (wxpy.Message), 18
img_width (wxpy.Message), 18
is_at (wxpy.Message), 18
is_friend (wxpy.User), 15
is_owner (wxpy.Group), 16

L

location (wxpy.Message), 18

logout() (wxpy.Bot), 13

M

member (wxpy.Message), 18
members (wxpy.Group), 15
mps() (wxpy.Bot), 12

N

name (wxpy.Chat), 14
nick_name (wxpy.Chat), 14

O

owner (wxpy.Group), 16

P

pin() (wxpy.Chat), 15
play_length (wxpy.Message), 18
province (wxpy.User), 15

R

raw (wxpy.Chat), 14
raw (wxpy.Message), 17
receiver (wxpy.Message), 18
register() (wxpy.Bot), 19
remark_name (wxpy.User), 15
remove_members() (wxpy.Group), 16
rename_group() (wxpy.Group), 16
reply() (wxpy.Message), 18
reply_file() (wxpy.Message), 18
reply_image() (wxpy.Message), 18
reply_msg() (wxpy.Message), 18
reply_raw_msg() (wxpy.Message), 18
reply_text() (wxpy.Tuling), 23
reply_video() (wxpy.Message), 18

S

search() (wxpy.Chats), 16
search() (wxpy.Group), 15
search() (wxpy.Groups), 17
send() (wxpy.Chat), 14

send_file() (wxpy.Chat), 14
send_image() (wxpy.Chat), 14
send_msg() (wxpy.Chat), 15
send_raw_msg() (wxpy.Chat), 15
send_video() (wxpy.Chat), 14
sender (wxpy.Message), 18
sex (wxpy.User), 15
signature (wxpy.User), 15
start() (wxpy.Bot), 19
stats() (wxpy.Chats), 16
stats_text() (wxpy.Chats), 16

T

text (wxpy.Message), 17
type (wxpy.Message), 17

U

unpin() (wxpy.Chat), 15
update_group() (wxpy.Group), 16
url (wxpy.Message), 18
user_details() (wxpy.Bot), 13
user_name (wxpy.Chat), 14

V

voice_length (wxpy.Message), 18

W

wxid (wxpy.Chat), 14
wxpy (), 7, 11, 13, 17, 21, 23, 24