
WhaPy Documentation

Release 0.0.1

Efraim Rodrigues

Feb 19, 2018

Contents:

1	API Reference	1
1.1	WhaPy	1
1.2	Event Reference	2
1.3	Enumeration	3
1.4	Data Classes	3
2	Indices and tables	7

1.1 WhaPy

WhaPy is a Python API for whatsapp. WhaPy listens to events from the app and dispatch them into beautiful coroutines.

Moreover, WhaPy module provides classes for managing whatsapp chats and messages.

class `whapy.WhaPy` (*browser, headless, loop=None*)

Represents the current whatsapp session.

This class is intended to give a python interface to interact with whatsapp web.

Parameters

- **browser** – A enumeration of the type *Browser*.
- **headless** – A boolean indicating whether the browser should run on headless mode or not.

get_me ()

Returns the logged in user's id

get_qrcode ()

Returns the image source of the qr code. If qr code is not visible due inactivity, it will be reloaded automatically

reload_qrcode ()

Clicks on reload qr code button to reload the once hidden qr code

event (*coro*)

A decorator that registers an event to listen to.

Example

Using the basic *event* () decorator:

```
@whatsapp.event
@asyncio.coroutine
def on_message(chat, messages):
    print('Received a message!')
```

run()

Starts listening to events

has_unread()

Returns True whenever there are unread messages

get_unread_chats()

Returns an array of unread chats' id

get_chats_id()

Returns an array of all chats' id. Use instantiateChat to have access to *Chat* methods

instantiate_chat(chatId)

Returns a *Chat* class

Parameters chatId – Chat identification

set_status(status)

Sets bot status :param status: A string containing the new status

leave_group(chatId)

Leaves a group :param chatId: Chat identification

1.2 Event Reference

This page outlines the different types of events listened by *WhaPy*.

This is an example of how to register an event:

```
from WhaPy import WhaPy

@asyncio.coroutine
async def on_message(chat, messages):
    for i in range(0, len(messages)):
        if not messages[i].is_media():
            print(messages[i].get_content())
            if messages[i].get_content() == "@who":
                chat.send_message("are you?")
```

Warning: All the events must be a *coroutine*. If they aren't, then you might get unexpected errors. In order to turn a function into a coroutine they must either be decorated with `@asyncio.coroutine` or in Python 3.5+ be defined using the `async def` declaration.

The following two functions are examples of coroutine functions:

```
async def on_ready():
    pass

@asyncio.coroutine
def on_ready():
    pass
```

`whapy.on_ready()`

Triggered after scanning QR Code and whatsapp page is ready. Chats and messages are filled up.

`whapy.on_message(chat, messages)`

Triggered each time unread messages are detected.

Parameters

- **chat** – A *Chat* of the chat which fired the event.
- **messages** – An array of *Message*.

1.3 Enumeration

The API provides a enumeration for browser selection.

All enumerations are subclasses of `enum`.

class `whapy.Browser`

Specifies the browser application will execute on.

Note: Drivers are required to interface with the chosen browser. Make sure it's in your PATH, e. g., place it in `/usr/bin` or `/usr/local/bin`.

Drivers' information and download links: <http://selenium-python.readthedocs.io/installation.html#drivers>

chrome

Google Chrome browser.

edge

Microsoft Edge.

firefox

Mozilla Firefox.

safari

Safari browser.

1.4 Data Classes

Classes that represent chats and messages in whatsapp web.

1.4.1 Chat

class `whapy.Chat`

Represents a whatsapp chat and allows chat management.

send_message (*message*)

Sends a message to this chat

Parameters **message** – A str to be sent to this chat

get_title ()

Returns the title of the chat

count_unread()

Returns the number of unread messages in this chat

count_loaded_msgs()

Returns the number of loaded messages in this chat

get_last_n_messages(*number*)

Returns the last N messages

Parameters **number** – int referring to the number of messages to retrieved

load_earlier_msgs()

Not all chat's messages are loaded at once. They're loaded in batches, so this will ask for more messages.

This will not return anything, but allow more messages to be retrieved

viewed()

Returns true if the chat has been viewed

set_pin(*boolean*)

Pins or unpins a chat according to the parameter

Parameters **boolean** – True for pinning and False otherwise

is_group()

Returns true if the chat is a group

mark_seen()

Mark the chat as seen

mark_unseen()

Mark the chat as unseen

is_mute()

Returns true is the chat is muted

mute(*time*)

Mute the chat for a certain time

Parameters **time** – Time for keeping chat muted

unmute()

Unmutes the chat

send_contact(*contactId*)

Sends the specified contact to the chat

Parameters **contactId** – Contact identification to be sent

delete()

Deletes messages but doesn't leave group

report_spam()

This will tag chat as spam. User will leave and delete chat. All messages will be deleted

1.4.2 Message

class whapy.**Message**

Represents a whatsapp message.

get_content()

Returns the content of the message

get_author()

Returns the author (contactId) of the message

get_from()

Returns the author (contactId) of the message. If message was sent in a group, the author is the group chat id

is_group_message()

Returns True if group message, False otherwise

is_link()

Returns True if message is a link, False otherwise

is_media()

Returns True if message is a media, False otherwise

is_quoted_msg()

Returns True if message is a quoted message, False otherwise

get_quoted_msg()

Returns the actually quoted message

get_quoted_participant()

Returns the author (contact id) of the quoted message

CHAPTER 2

Indices and tables

- `genindex`
- `search`

B

Browser (class in whapy), 3

C

Chat (class in whapy), 3

chrome (whapy.Browser attribute), 3

count_loaded_msgs() (whapy.Chat method), 4

count_unread() (whapy.Chat method), 3

D

delete() (whapy.Chat method), 4

E

edge (whapy.Browser attribute), 3

event() (whapy.WhaPy method), 1

F

firefox (whapy.Browser attribute), 3

G

get_author() (whapy.Message method), 4

get_chats_id() (whapy.WhaPy method), 2

get_content() (whapy.Message method), 4

get_from() (whapy.Message method), 5

get_last_n_messages() (whapy.Chat method), 4

get_me() (whapy.WhaPy method), 1

get_qrcode() (whapy.WhaPy method), 1

get_quoted_msg() (whapy.Message method), 5

get_quoted_participant() (whapy.Message method), 5

get_title() (whapy.Chat method), 3

get_unread_chats() (whapy.WhaPy method), 2

H

has_unread() (whapy.WhaPy method), 2

I

instantiate_chat() (whapy.WhaPy method), 2

is_group() (whapy.Chat method), 4

is_group_message() (whapy.Message method), 5

is_link() (whapy.Message method), 5

is_media() (whapy.Message method), 5

is_mute() (whapy.Chat method), 4

is_quoted_msg() (whapy.Message method), 5

L

leave_group() (whapy.WhaPy method), 2

load_earlier_msgs() (whapy.Chat method), 4

M

mark_seen() (whapy.Chat method), 4

mark_unseen() (whapy.Chat method), 4

Message (class in whapy), 4

mute() (whapy.Chat method), 4

O

on_message() (in module whapy), 3

on_ready() (in module whapy), 2

R

reload_qrcode() (whapy.WhaPy method), 1

report_spam() (whapy.Chat method), 4

run() (whapy.WhaPy method), 2

S

safari (whapy.Browser attribute), 3

send_contact() (whapy.Chat method), 4

send_message() (whapy.Chat method), 3

set_pin() (whapy.Chat method), 4

set_status() (whapy.WhaPy method), 2

U

unmute() (whapy.Chat method), 4

V

viewed() (whapy.Chat method), 4

W

WhaPy (class in whapy), 1