

---

# **PyTox Documentation**

*Release 0.0.23*

**Wei-Ning Huang**

January 15, 2016



## CONTENTS

<b>1 About</b>	<b>1</b>
<b>2 Module Documentation</b>	<b>3</b>
<b>3 Class Documentation</b>	<b>9</b>
<b>4 Indices and tables</b>	<b>11</b>
<b>Index</b>	<b>13</b>



## ABOUT

PyTox is the official Python binding for [Project-Tox](#).

The project page/repo is located at [Github](#).



## MODULE DOCUMENTATION

### 2.1 pytox Module

The pytox module contains two classes:

#### 2.1.1 pytox.Tox Class

PyTox wraps libtoxcore API with the object `pytox.Tox`, all the tox API methods are in this class.

**class** `pytox.Tox`

ToxCore object

**add\_groupchat** ()

Creates a new groupchat and puts it in the chats array.

**add\_tcp\_relay** (*address, port, public\_key*)

**bootstrap** (*address, port, public\_key*)

Resolves address into an IP address. If successful, sends a 'get nodes' request to the given node with ip, port.

**count\_chatlist** ()

Return the number of chats in the current Tox instance.

**del\_groupchat** (*group\_number*)

Delete a groupchat from the chats array.

**file\_control** (*friend\_number, file\_number, control*)

Send a file send request. Returns file number to be sent.

**file\_get\_file\_id** (*friend\_number, file\_number*)

Send a file send request. Returns file id's hex string

**file\_seek** (*friend\_number, file\_number, position*)

Send a file send request. Returns file number to be sent.

**file\_send** (*friend\_number, kind, file\_size, file\_id, filename*)

Send a file send request. Returns file number to be sent.

**file\_send\_chunk** (*friend\_number, file\_number, position, data*)

Send a file send request. Returns file number to be sent.

**friend\_add** (*address, message*)

Add a friend.

**friend\_add\_norequest** (*address*)

Add a friend without sending request.

**friend\_by\_public\_key** (*friend\_id*)

Return the friend id associated to that client id.

**friend\_delete** (*friend\_number*)

Remove a friend.

**friend\_exists** (*friend\_number*)

Checks if there exists a friend with given friendnumber.

**friend\_get\_connection\_status** (*friend\_number*)

Return True if friend is connected(Online) else False.

**friend\_get\_last\_online** (*friend\_number*)

returns datetime.datetime object representing the last time *friend\_number* was seen online, or None if never seen.

**friend\_get\_name** (*friend\_number*)

Get nickname of *friend\_number*.

**friend\_get\_name\_size** (*friend\_number*)

Get nickname length of *friend\_number*.

**friend\_get\_public\_key** (*friend\_number*)

Return the public key associated to that friend number.

**friend\_get\_status** (*friend\_number*)

Get friend status.

**See also:**

`set_user_status()`

**friend\_get\_status\_message** (*friend\_number*)

Get status message of a friend.

**friend\_get\_status\_message\_size** (*friend\_number*)

Return the length of *friend\_number*'s status message.

**friend\_get\_typing** (*friend\_number*)

Return True is user is typing.

**friend\_send\_message** (*friend\_number, type, message*)

Send a text chat message to an online friend.

**get\_chatlist** ()

Return a list of valid group numbers.

**get\_savedata** ()

Return messenger blob in str.

**get\_savedata\_size** ()

return size of messenger data (for saving).

**group\_action\_send** (*group\_number, action*)

send a group action.

**group\_get\_names** (*group\_number*)

List all the peers in the group chat.

**group\_get\_title** (*group\_number*)

Returns the title for *group\_number*.

**group\_get\_type** (*group\_number*)

Return the type of group, could be the following value:

type	description
Tox.GROUPCHAT_TYPE_TEXT	text chat
Tox.GROUPCHAT_TYPE_AV	video chat

**group\_message\_send** (*group\_number, message*)  
send a group message.

**group\_number\_peers** (*group\_number*)  
Return the number of peers in the group chat.

**group\_peername** (*group\_number, peer\_number*)  
Get the group peer's name.

**group\_set\_title** (*group\_number, title*)  
Sets the title for group.

**invite\_friend** (*friend\_number, group\_number*)  
Invite friendnumber to groupnumber.

**iterate** ()  
The main loop that needs to be run at least 20 times per second.

**iteration\_interval** ()  
returns time (in ms) before the next tox\_iterate() needs to be run on success.

**join\_groupchat** (*friend\_number, data*)  
Join a group (you need to have been invited first.). Returns the group number of success.

**kill** ()  
Run this before closing shop.

**on\_file\_chunk\_request** (*friend\_number, file\_number, position, length*)  
Callback for more file chunk, default implementation does nothing.

**on\_file\_recv** (*friend\_number, file\_number, kind, file\_size, filename*)  
Callback for receiving file transfer, default implementation does nothing.

**on\_file\_recv\_chunk** (*friend\_number, file\_number, position, data*)  
Callback for receiving file chunk, default implementation does nothing.

**on\_file\_recv\_control** (*friend\_number, file\_number, control*)  
Callback for receiving file control, default implementation does nothing.

**on\_friend\_connection\_status** (*friend\_number, status*)  
Callback for receiving read receipt, default implementation does nothing.

*status* is a boolean value which indicates the status of the friend indicated by *friend\_number*. True means online and False means offline after previously online.

**on\_friend\_message** (*friend\_number, type, message*)  
Callback for receiving friend messages, default implementation does nothing.

**on\_friend\_name** (*friend\_number, new\_name*)  
Callback for receiving friend name changes, default implementation does nothing.

**on\_friend\_read\_receipt** (*friend\_number, receipt*)  
Callback for receiving read receipt, default implementation does nothing.

**on\_friend\_request** (*address, message*)  
Callback for receiving friend requests, default implementation does nothing.

**on\_friend\_status** (*friend\_number, kind*)  
Callback for receiving friend status changes, default implementation does nothing.

**on\_friend\_status\_message** (*friend\_number, new\_status*)

Callback for receiving friend status message changes, default implementation does nothing.

**on\_friend\_typing** (*friend\_number, is\_typing*)

Callback for receiving friend status changes, default implementation does nothing.

**on\_group\_action** (*group\_number, friend\_group\_number, action*)

Callback for receiving group actions, default implementation does nothing.

**on\_group\_invite** (*friend\_number, type, group\_public\_key*)

Callback for receiving group invitations, default implementation does nothing.

**See also:**

*group\_get\_type()*

**on\_group\_message** (*group\_number, friend\_group\_number, message*)

Callback for receiving group messages, default implementation does nothing.

**on\_group\_namelist\_change** (*group\_number, peer\_number, change*)

Callback for receiving group messages, default implementation does nothing.

There are there possible *change* values:

change	description
Tox.CHAT_CHANGE_PEER_ADD	a peer is added
Tox.CHAT_CHANGE_PEER_DEL	a peer is deleted
Tox.CHAT_CHANGE_PEER_NAME	name of peer changed

**on\_self\_connection\_status** (*friend\_number, status*)

Callback for receiving read receipt, default implementation does nothing.

**self\_get\_address** ()

Return address to give to others.

**self\_get\_connection\_status** ()

Return False if we are not connected to the DHT.

**self\_get\_friend\_list** ()

Get a list of valid friend numbers.

**self\_get\_friend\_list\_size** ()

Return the number of friends.

**self\_get\_keys** ()

Get the public and secret key from the Tox object. Return a tuple (public\_key, secret\_key)

**self\_get\_name** ()

Get our self nickname.

**self\_get\_name\_size** ()

Get our self nickname string length

**self\_get\_nospam** ()

get nospam part from ID

**self\_get\_status** ()

Get user status of yourself.

**See also:**

*set\_user\_status()*

**self\_get\_status\_message** ()

Get status message of yourself.

**self\_get\_status\_message\_size** ()  
Get status message string length of yourself.

**self\_set\_name** (*name*)  
Set our self nickname.

**self\_set\_nospam** (*nospam*)  
set nospam part of ID. *nospam* should be of type uint32

**self\_set\_status** (*status*)  
Set our user status, status can have following values:

kind	description
Tox.USERSTATUS_NONE	the user is online
Tox.USERSTATUS_AWAY	the user is away
Tox.USERSTATUS_BUSY	the user is busy
Tox.USERSTATUS_INVALID	invalid status

**self\_set\_status\_message** (*message*)  
Set our self status message.

**self\_set\_typing** (*friend\_number, is\_typing*)  
Set user typing status.

## 2.1.2 pytox.ToxAV Class

PyTox wraps libtoxav API with the object pytox.ToxAV, all the toxav API methods are in this class.

```
class pytox.ToxAV
    ToxAV object

    answer (friend_number, audio_bit_rate, video_bit_rate)
        Answer incomming call.

    See also:
        call ()

    audio_send_frame (friend_number, pcm, sample_count, channels, sampling_rate)
        Send an audio frame to a friend.Returns True on success.

    bit_rate_set (friend_number, audio_bit_rate, video_bit_rate)
        Set the bit rate to be used in subsequent audio/video frames.Returns True on success.

    call (friend_number, audio_bit_rate, video_bit_rate)
        Call a friend with friend_number.Returns True on success.

    call_control (friend_number, control)
        Sends a call control command to a friend.Returns True on success.

    get_tox ()
        Get the Tox object associated with this ToxAV instance.

    group_send_audio (groupnumber, pcm, samples, channels, sample_rate)
        Send audio to the group chat.Returns -1 on failure.

    iterate ()
        Main loop for the session.

    iteration_interval ()
        Returns the interval in milliseconds when the next toxav_iterate call shouldbe.
```

**join\_av\_groupchat** (*friend\_number, data*)

Join a AV group (you need to have been invited first.)Returns -1 on failure.

**video\_send\_frame** (*friend\_number, width, height, y, u, v*)

Send a video frame to a friend.Returns True on success.

**CLASS DOCUMENTATION**



## INDICES AND TABLES

- genindex
- modindex
- search



**A**

add\_groupchat() (pytox.Tox method), 3  
 add\_tcp\_relay() (pytox.Tox method), 3  
 answer() (pytox.ToxAV method), 7  
 audio\_send\_frame() (pytox.ToxAV method), 7

**B**

bit\_rate\_set() (pytox.ToxAV method), 7  
 bootstrap() (pytox.Tox method), 3

**C**

call() (pytox.ToxAV method), 7  
 call\_control() (pytox.ToxAV method), 7  
 count\_chatlist() (pytox.Tox method), 3

**D**

del\_groupchat() (pytox.Tox method), 3

**F**

file\_control() (pytox.Tox method), 3  
 file\_get\_file\_id() (pytox.Tox method), 3  
 file\_seek() (pytox.Tox method), 3  
 file\_send() (pytox.Tox method), 3  
 file\_send\_chunk() (pytox.Tox method), 3  
 friend\_add() (pytox.Tox method), 3  
 friend\_add\_norequest() (pytox.Tox method), 3  
 friend\_by\_public\_key() (pytox.Tox method), 4  
 friend\_delete() (pytox.Tox method), 4  
 friend\_exists() (pytox.Tox method), 4  
 friend\_get\_connection\_status() (pytox.Tox method), 4  
 friend\_get\_last\_online() (pytox.Tox method), 4  
 friend\_get\_name() (pytox.Tox method), 4  
 friend\_get\_name\_size() (pytox.Tox method), 4  
 friend\_get\_public\_key() (pytox.Tox method), 4  
 friend\_get\_status() (pytox.Tox method), 4  
 friend\_get\_status\_message() (pytox.Tox method), 4  
 friend\_get\_status\_message\_size() (pytox.Tox method), 4  
 friend\_get\_typing() (pytox.Tox method), 4  
 friend\_send\_message() (pytox.Tox method), 4

**G**

get\_chatlist() (pytox.Tox method), 4

get\_savedata() (pytox.Tox method), 4  
 get\_savedata\_size() (pytox.Tox method), 4  
 get\_tox() (pytox.ToxAV method), 7  
 group\_action\_send() (pytox.Tox method), 4  
 group\_get\_names() (pytox.Tox method), 4  
 group\_get\_title() (pytox.Tox method), 4  
 group\_get\_type() (pytox.Tox method), 4  
 group\_message\_send() (pytox.Tox method), 5  
 group\_number\_peers() (pytox.Tox method), 5  
 group\_peername() (pytox.Tox method), 5  
 group\_send\_audio() (pytox.ToxAV method), 7  
 group\_set\_title() (pytox.Tox method), 5

**I**

invite\_friend() (pytox.Tox method), 5  
 iterate() (pytox.Tox method), 5  
 iterate() (pytox.ToxAV method), 7  
 iteration\_interval() (pytox.Tox method), 5  
 iteration\_interval() (pytox.ToxAV method), 7

**J**

join\_av\_groupchat() (pytox.ToxAV method), 7  
 join\_groupchat() (pytox.Tox method), 5

**K**

kill() (pytox.Tox method), 5

**O**

on\_file\_chunk\_request() (pytox.Tox method), 5  
 on\_file\_recv() (pytox.Tox method), 5  
 on\_file\_recv\_chunk() (pytox.Tox method), 5  
 on\_file\_recv\_control() (pytox.Tox method), 5  
 on\_friend\_connection\_status() (pytox.Tox method), 5  
 on\_friend\_message() (pytox.Tox method), 5  
 on\_friend\_name() (pytox.Tox method), 5  
 on\_friend\_read\_receipt() (pytox.Tox method), 5  
 on\_friend\_request() (pytox.Tox method), 5  
 on\_friend\_status() (pytox.Tox method), 5  
 on\_friend\_status\_message() (pytox.Tox method), 5  
 on\_friend\_typing() (pytox.Tox method), 6  
 on\_group\_action() (pytox.Tox method), 6  
 on\_group\_invite() (pytox.Tox method), 6

on\_group\_message() (pytox.Tox method), 6  
on\_group\_namelist\_change() (pytox.Tox method), 6  
on\_self\_connection\_status() (pytox.Tox method), 6

## S

self\_get\_address() (pytox.Tox method), 6  
self\_get\_connection\_status() (pytox.Tox method), 6  
self\_get\_friend\_list() (pytox.Tox method), 6  
self\_get\_friend\_list\_size() (pytox.Tox method), 6  
self\_get\_keys() (pytox.Tox method), 6  
self\_get\_name() (pytox.Tox method), 6  
self\_get\_name\_size() (pytox.Tox method), 6  
self\_get\_nospam() (pytox.Tox method), 6  
self\_get\_status() (pytox.Tox method), 6  
self\_get\_status\_message() (pytox.Tox method), 6  
self\_get\_status\_message\_size() (pytox.Tox method), 6  
self\_set\_name() (pytox.Tox method), 7  
self\_set\_nospam() (pytox.Tox method), 7  
self\_set\_status() (pytox.Tox method), 7  
self\_set\_status\_message() (pytox.Tox method), 7  
self\_set\_typing() (pytox.Tox method), 7

## T

Tox (class in pytox), 3  
ToxAV (class in pytox), 7

## V

video\_send\_frame() (pytox.ToxAV method), 8