
twx.botapi Documentation

Release 1.0

Vince Castellano, Phillip Lopo

July 04, 2015

1	twx.botapi — Unofficial Telegram Bot API Client	3
1.1	Telegram Bot API Types	3
1.2	Additional Types	11
1.3	Request Objects	12
1.4	Telegram Bot API Methods	13
2	Indices and tables	19
	Python Module Index	21

Contents:

twx.botapi — Unofficial Telegram Bot API Client

1.1 Telegram Bot API Types

1.1.1 User

class `twx.botapi.User`

This object represents a Telegram user or bot.

id

int

Unique identifier for this user or bot

first_name

str

User's or bot's first name

last_name

Optional[str]

User's or bot's last name

username

Optional[str]

User's or bot's username

1.1.2 GroupChat

class `twx.botapi.GroupChat`

This object represents a group chat.

id

int

Unique identifier for this group chat

title

str

Group name

1.1.3 Message

class `twx.botapi.Message`

This object represents a message.

message_id

int

Unique message identifier

from

User

Sender

date

int

Date the message was sent in Unix time

chat

User or GroupChat

Conversation the message belongs to — user in case of a private message, GroupChat in case of a group

forward_from

User

Optional. For forwarded messages, sender of the original message

forward_date

int

Optional. For forwarded messages, date the original message was sent in Unix time

reply_to_message

Message

Optional. For replies, the original message. Note that the Message object in this field will not contain further reply_to_message fields even if it itself is a reply.

text

str

Optional. For text messages, the actual UTF-8 text of the message

audio

Audio

Optional. Message is an audio file, information about the file

document

Document

Optional. Message is a general file, information about the file

photo

Sequence[PhotoSize]

Optional. Message is a photo, available sizes of the photo

sticker

Sticker

Optional. Message is a sticker, information about the sticker

video*Video**Optional.* Message is a video, information about the video**contact***Contact**Optional.* Message is a shared contact, information about the contact**location***Location**Optional.* Message is a shared location, information about the location**new_chat_participant***User**Optional.* A new member was added to the group, information about them (this member may be bot itself)**left_chat_participant***User**Optional.* A member was removed from the group, information about them (this member may be bot itself)**new_chat_title***str**Optional.* A group title was changed to this value**new_chat_photo***Sequence[PhotoSize]**Optional.* A group photo was change to this value**delete_chat_photo***True**Optional.* Informs that the group photo was deleted**group_chat_created***True**Optional.* Informs that the group has been created

1.1.4 PhotoSize

class twx.botapi.PhotoSize

This object represents one size of a photo or a file / sticker thumbnail.

file_id*str*

Unique identifier for this file

width*int*

Photo width

height*int*

Photo height

file_size

int

Optional. File size

1.1.5 Audio

class `twx.botapi.Audio`

This object represents an audio file (voice note).

file_id

str

Unique identifier for this file

duration

int

Duration of the audio in seconds as defined by sender

mime_type

str

Optional. MIME type of the file as defined by sender

file_size

int

Optional. File size

1.1.6 Document

class `twx.botapi.Document`

This object represents a general file (as opposed to photos and audio files).

file_id

str

Unique file identifier

thumb

PhotoSize

Document thumbnail as defined by sender

file_name

str

Optional. Original filename as defined by sender

mime_type

str

Optional. MIME type of the file as defined by sender

file_size

int

Optional. File size

1.1.7 Sticker

class `twx.botapi.Sticker`

This object represents a sticker.

file_id

str

Unique identifier for this file

width

int

Sticker width

height

int

Sticker height

thumb

PhotoSize

Sticker thumbnail in .webp or .jpg format

file_size

int

Optional. File size

1.1.8 Video

class `twx.botapi.Video`

This object represents a video file.

file_id

str

Unique identifier for this file

width

int

Video width as defined by sender

height

int

Video height as defined by sender

duration

int

Duration of the video in seconds as defined by sender

thumb

PhotoSize

Video thumbnail

mime_type

str

Optional. Mime type of a file as defined by sender

file_size

int

Optional. File size

caption

str

Optional. Text description of the video (usually empty)

1.1.9 Contact

class `twx.botapi.Contact`

This object represents a phone contact.

phone_number

str

Contact's phone number

first_name

str

Contact's first name

last_name

str

Optional. Contact's last name

user_id

str

Optional. Contact's user identifier in Telegram

1.1.10 Location

class `twx.botapi.Location`

This object represents a point on the map.

longitude

float

Longitude as defined by sender

latitude

float

Latitude as defined by sender

1.1.11 Update

class `twx.botapi.Update`

This object represents an incoming update.

update_id

int

The update's unique identifier. Update identifiers start from a certain positive number and increase sequentially. This ID becomes especially handy if you're using Webhooks, since it allows you to ignore repeated updates or to restore the correct update sequence, should they get out of order.

message

Message

Optional. New incoming message of any kind — text, photo, sticker, etc.

1.1.12 InputFile

class twx.botapi.**InputFile**

This object represents the contents of a file to be uploaded. Must be posted using multipart/form-data in the usual way that files are uploaded via the browser.

form

str

the form used to submit (e.g. 'photo')

file_info

InputFileInfo

The file metadata required

Example

```
fp = open('foo.png', 'rb')
file_info = InputFileInfo('foo.png', fp, 'image/png')

InputFile('photo', file_info)

bot.send_photo(chat_id=12345678, photo=InputFile)
```

Note: While creating the FileInput currently requires a reasonable amount of preparation just to send a file. This class will be extended in the future to make the process easier.

1.1.13 UserProfilePhotos

class twx.botapi.**UserProfilePhotos**

This object represent a user's profile pictures.

total_count

int

Total number of profile pictures the target user has

photos

list of list of PhotoSize

Requested profile pictures (in up to 4 sizes each)

1.1.14 ReplyKeyboardMarkup

class `twx.botapi.ReplyKeyboardMarkup`

This object represents a custom keyboard with reply options (see Introduction to bots for details and examples).

keyboard

list of list of str

Array of button rows, each represented by an Array of Strings

resize_keyboard

bool

Optional. Requests clients to resize the keyboard vertically for optimal fit (e.g., make the keyboard smaller if there are just two rows of buttons). Defaults to false, in which case the custom keyboard is always of the same height as the app's standard keyboard.

one_time_keyboard

bool

Optional. Requests clients to hide the keyboard as soon as it's been used. Defaults to false.

selective

bool

Optional. Use this parameter if you want to show the keyboard to specific users only. Targets:

- 1.users that are @mentioned in the text of the Message object;
- 2.if the bot's message is a reply (has `reply_to_message_id`), sender of the original message.

Example A user requests to change the bot's language, bot replies to the request with a keyboard to select the new language. Other users in the group don't see the keyboard.

Example

```
keyboard = [
    ['7', '8', '9'],
    ['4', '5', '6'],
    ['1', '2', '3'],
    ['0']]

reply_markup = ReplyKeyboardMarkup.create(keyboard)
bot.send_message(12345678, 'testing reply_markup', reply_markup=reply_markup)
```

1.1.15 ReplyKeyboardHide

class `twx.botapi.ReplyKeyboardHide`

Upon receiving a message with this object, Telegram clients will hide the current custom keyboard and display the default letter-keyboard. By default, custom keyboards are displayed until a new keyboard is sent by a bot. An exception is made for one-time keyboards that are hidden immediately after the user presses a button (see [ReplyKeyboardMarkup](#)).

hide_keyboard

True

Requests clients to hide the custom keyboard

selective

bool

Optional. Use this parameter if you want to hide keyboard for specific users only. Targets:

1. users that are @mentioned in the text of the Message object;
2. if the bot's message is a reply (has `reply_to_message_id`), sender of the original message.

Example A user votes in a poll, bot returns confirmation message in reply to the vote and hides keyboard for that user, while still showing the keyboard with poll options to users who haven't voted yet.

1.1.16 ForceReply

class `twx.botapi.ForceReply`

Upon receiving a message with this object, Telegram clients will display a reply interface to the user (act as if the user has selected the bot's message and tapped 'Reply'). This can be extremely useful if you want to create user-friendly step-by-step interfaces without having to sacrifice privacy mode.

force_reply

True

Shows reply interface to the user, as if they manually selected the bot's message and tapped 'Reply'

selective

bool

Optional. Use this parameter if you want to force reply from specific users only. Targets: 1) users that are @mentioned in the text of the Message object; 2) if the bot's message is a reply (has `reply_to_message_id`), sender of the original message.

Example A poll bot for groups runs in privacy mode (only receives commands, replies to its messages and mentions). There could be two ways to create a new poll:

- Explain the user how to send a command with parameters (e.g. `/newpoll question answer1 answer2`). May be appealing for hardcore users but lacks modern day polish.
- Guide the user through a step-by-step process. 'Please send me your question', 'Cool, now let's add the first answer option', 'Great. Keep adding answer options, then send `/done` when you're ready'.

The last option is definitely more attractive. And if you use `ForceReply` in your bot's questions, it will receive the user's answers even if it only receives replies, commands and mentions — without any extra work for the user.

1.2 Additional Types

1.2.1 Error

class `twx.botapi.Error`

The error code and message returned when a request was successfully but the method call was invalid

error_code

int

An Integer 'error_code' field is also returned, but its contents are subject to change in the future.

description

str

The description of the error as reported by Telegram

1.3 Request Objects

class `twx.botapi.RequestMethod`

Used to specify the HTTP request method.

GET

'GET'

POST

'POST'

Example

```
bot.get_me(request_method=RequestMethod.GET)
```

class `twx.botapi.TelegramBotRPCRequest` (*api_method*, *token*, *params=None*, *on_result=None*, *on_success=None*, *callback=None*, *on_error=None*, *files=None*, *request_method=<RequestMethod.POST: 'POST'>*)

Class that handles creating the actual RPC request, and sending callbacks based on response

Parameters

- **api_method** (*str*) – The API method to call. See <https://core.telegram.org/bots/api#available-methods>
- **token** (*str*) – The API token generated following the instructions at <https://core.telegram.org/bots#botfather>
- **params** (*dict*) – A dictionary mapping the api method parameters to their arguments
- **on_result** (*callable*) – a callback function that gets called before when the request finishes. The return value of this function gets passed to `on_success`. Useful if you wish to override the generated result type
- **on_success** (*callable*) – a callback function that gets called when the api call was successful, gets passed the return value from `on_result`
- **on_error** (*callable*) – called when an error occurs
- **files** (*list of InputFile*) – a list of `InputFile`'s to be sent to the server
- **request_method** (*RequestMethod*) – `RequestMethod.POST` or `RequestMethod.GET`

Note: Typically you do not have to interact with this class directly. However, you may override any of these arguments by specifying it in the `**kwargs` of any api method.

1.4 Telegram Bot API Methods

1.4.1 get_me

`twx.botapi.get_me(*, request_args=None, **kwargs)`

A simple method for testing your bot's auth token. Requires no parameters. Returns basic information about the bot in form of a `User` object.

Parameters ****kwargs** – Args that get passed down to `TelegramBotRPCRequest`

Returns Returns basic information about the bot in form of a `User` object.

Return type `User`

1.4.2 send_message

`twx.botapi.send_message(chat_id, text, disable_web_page_preview=None, reply_to_message_id=None, reply_markup=*, request_args=None, **kwargs)`

Use this method to send text messages.

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **text** (*str*) – Text of the message to be sent
- **disable_web_page_preview** (*bool*) – Disables link previews for links in this message
- **reply_to_message_id** (*int*) – If the message is a reply, ID of the original message
- **reply_markup** (`ReplyKeyboardMarkup` or `ReplyKeyboardHide` or `ForceReply`) – Additional interface options. A JSON-serialized object for a custom reply keyboard, instructions to hide keyboard or to force a reply from the user.
- ****kwargs** – Args that get passed down to `TelegramBotRPCRequest`

Returns On success, the sent `Message` is returned.

Return type `Message`

1.4.3 forward_message

`twx.botapi.forward_message(chat_id, from_chat_id, message_id, *, request_args=None, **kwargs)`

Use this method to forward messages of any kind.

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **from_chat_id** (*int*) – Unique identifier for the chat where the original message was sent — User or GroupChat id
- **message_id** (*int*) – Unique message identifier
- ****kwargs** – Args that get passed down to `TelegramBotRPCRequest`

Returns On success, the sent `Message` is returned.

Return type `Message`

1.4.4 send_photo

`twx.botapi.send_photo(chat_id, photo, caption=None, reply_to_message_id=None, reply_markup=*, request_args=None, **kwargs)`

Use this method to send photos.

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **photo** (*InputFile or str*) – Photo to send. You can either pass a file_id as String to resend a photo that is already on the Telegram servers, or upload a new photo using multipart/form-data.
- **caption** (*str*) – Photo caption (may also be used when resending photos by file_id).
- **reply_to_message_id** (*int*) – If the message is a reply, ID of the original message
- **reply_markup** (*ReplyKeyboardMarkup or ReplyKeyboardHide or ForceReply*) – Additional interface options. A JSON-serialized object for a custom reply keyboard, instructions to hide keyboard or to force a reply from the user.
- ****kwargs** – Args that get passed down to *TelegramBotRPCRequest*

Returns On success, the sent Message is returned.

Return type *TelegramBotRPCRequest*

1.4.5 send_audio

`twx.botapi.send_audio(chat_id, audio, reply_to_message_id=None, reply_markup=None, *, request_args=None, **kwargs)`

Use this method to send audio files, if you want Telegram clients to display the file as a playable voice message. For this to work, your audio must be in an .ogg file encoded with OPUS (other formats may be sent as Document).

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **audio** (*InputFile or str*) – Audio file to send. You can either pass a file_id as String to resend an audio that is already on the Telegram servers, or upload a new audio file using multipart/form-data.
- **reply_to_message_id** (*int*) – If the message is a reply, ID of the original message
- **reply_markup** (*ReplyKeyboardMarkup or ReplyKeyboardHide or ForceReply*) – Additional interface options. A JSON-serialized object for a custom reply keyboard,
- ****kwargs** – Args that get passed down to *TelegramBotRPCRequest*

Returns On success, the sent Message is returned.

Return type *TelegramBotRPCRequest*

1.4.6 send_document

`twx.botapi.send_document(chat_id, document, reply_to_message_id=None, reply_markup=None, *, request_args=None, **kwargs)`

Use this method to send general files.

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **document** (*InputFile or str*) – File to send. You can either pass a file_id as String to resend a file that is already on the Telegram servers, or upload a new file using multipart/form-data.
- **reply_to_message_id** (*int*) – If the message is a reply, ID of the original message
- **reply_markup** (*ReplyKeyboardMarkup or ReplyKeyboardHide or ForceReply*) – Additional interface options. A JSON-serialized object for a custom reply keyboard, instructions to hide keyboard or to force a reply from the user.
- ****kwargs** – Args that get passed down to *TelegramBotRPCRequest*

Returns On success, the sent Message is returned.

Return type *TelegramBotRPCRequest*

1.4.7 send_sticker

`twx.botapi.send_sticker(chat_id, sticker, reply_to_message_id=None, reply_markup=None, *, request_args=None, **kwargs)`

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **sticker** (*InputFile or str*) – Sticker to send. You can either pass a file_id as String to resend a sticker that is already on the Telegram servers, or upload a new sticker using multipart/form-data.
- **reply_to_message_id** (*int*) – If the message is a reply, ID of the original message
- **reply_markup** (*ReplyKeyboardMarkup or ReplyKeyboardHide or ForceReply*) – Additional interface options. A JSON-serialized object for a custom reply keyboard, instructions to hide keyboard or to force a reply from the user.
- ****kwargs** – Args that get passed down to *TelegramBotRPCRequest*

Returns On success, the sent Message is returned.

Return type *TelegramBotRPCRequest*

1.4.8 send_video

`twx.botapi.send_video(chat_id, video, reply_to_message_id=None, reply_markup=None, *, request_args=None, **kwargs)`

Use this method to send video files, Telegram clients support mp4 videos (other formats may be sent as Document).

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **video** (*InputFile or str*) – Video to send. You can either pass a file_id as String to resend a video that is already on the Telegram servers, or upload a new video using multipart/form-data.
- **reply_to_message_id** (*int*) – If the message is a reply, ID of the original message
- **reply_markup** (*ReplyKeyboardMarkup or ReplyKeyboardHide or ForceReply*) – Additional interface options. A JSON-serialized object for a custom reply keyboard, instructions to hide keyboard or to force a reply from the user.

- ****kwargs** – Args that get passed down to *TelegramBotRPCRequest*

Returns On success, the sent Message is returned.

Return type *TelegramBotRPCRequest*

1.4.9 send_location

```
twx.botapi.send_location(chat_id, latitude, longitude, reply_to_message_id=None, re-  
ply_markup=None, *, request_args=None, **kwargs)
```

Use this method to send point on the map.

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **latitude** (*float*) – Latitude of location.
- **longitude** (*float*) – Longitude of location.
- **reply_to_message_id** (*int*) – If the message is a reply, ID of the original message
- **reply_markup** (*ReplyKeyboardMarkup* or *ReplyKeyboardHide* or *ForceReply*) – Additional interface options. A JSON-serialized object for a custom reply keyboard, instructions to hide keyboard or to force a reply from the user.
- ****kwargs** – Args that get passed down to *TelegramBotRPCRequest*

Returns On success, the sent Message is returned.

Return type *TelegramBotRPCRequest*

1.4.10 send_chat_action

```
twx.botapi.send_chat_action(chat_id, action, *, request_args=None, **kwargs)
```

Use this method when you need to tell the user that something is happening on the bot's side. The status is set for 5 seconds or less (when a message arrives from your bot, Telegram clients clear its typing status).

Example: The ImageBot needs some time to process a request and upload the image. Instead of sending a text message along the lines of “Retrieving image, please wait...”, the bot may use `sendChatAction` with `action = upload_photo`. The user will see a “sending photo” status for the bot.

We only recommend using this method when a response from the bot will take a noticeable amount of time to arrive.

Parameters

- **chat_id** (*int*) – Unique identifier for the message recipient — User or GroupChat id
- **action** (*ChatAction*) – Type of action to broadcast. Choose one, depending on what the user is about to receive: `typing` for text messages, `upload_photo` for photos, `record_video` or `upload_video` for videos, `record_audio` or `upload_audio` for audio files, `upload_document` for general files, `find_location` for location data.
- ****kwargs** – Args that get passed down to *TelegramBotRPCRequest*

Returns Returns True on success.

Return type `bool`

1.4.11 get_user_profile_photos

`twx.botapi.get_user_profile_photos` (*user_id*, *offset=None*, *limit=None*, *, *request_args=None*, ***kwargs*)

Use this method to get a list of profile pictures for a user. Returns a `UserProfilePhotos` object.

Parameters

- **user_id** (*int*) – Unique identifier of the target user
- **offset** (*int*) – Sequential number of the first photo to be returned. By default, all photos are returned.
- **limit** (*int*) – Limits the number of photos to be retrieved. Values between 1—100 are accepted. Defaults to 100.
- ****kwargs** – Args that get passed down to `TelegramBotRPCRequest`

Returns Returns a `UserProfilePhotos` object.

Return type `TelegramBotRPCRequest`

1.4.12 get_updates

`twx.botapi.get_updates` (*offset=None*, *limit=None*, *timeout=None*, *, *request_args*, ***kwargs*)

Use this method to receive incoming updates using long polling.

Note:

- 1.This method will not work if an outgoing webhook is set up.
 - 2.In order to avoid getting duplicate updates, recalculate offset after each server response.
-

Parameters

- **offset** (*int*) – Identifier of the first update to be returned. Must be greater by one than the highest among the identifiers of previously received updates. By default, updates starting with the earliest unconfirmed update are returned. An update is considered confirmed as soon as `getUpdates` is called with an offset higher than its `update_id`.
- **limit** (*int*) – Limits the number of updates to be retrieved. Values between 1—100 are accepted. Defaults to 100
- **timeout** (*int*) – Timeout in seconds for long polling. Defaults to 0, i.e. usual short polling
- ****kwargs** – Args that get passed down to `TelegramBotRPCRequest`

Returns An Array of Update objects is returned.

Return type `TelegramBotRPCRequest`

1.4.13 set_webhook

`twx.botapi.set_webhook` (*url=None*, *, *request_args=None*, ***kwargs*)

Use this method to specify a url and receive incoming updates via an outgoing webhook. Whenever there is an update for the bot, we will send an HTTPS POST request to the specified url, containing a JSON-serialized Update. In case of an unsuccessful request, we will give up after a reasonable amount of attempts.

Please note that you will not be able to receive updates using `getUpdates` for as long as an outgoing webhook is set up.

Parameters

- **url** (*str*) – HTTPS url to send updates to. Use an empty string to remove webhook integration
- ****kwargs** – Args that get passed down to *TelegramBotRPCRequest*

Returns Returns True on success.

Return type *TelegramBotRPCRequest*

Indices and tables

- `genindex`
- `modindex`
- `search`

t

`twx.botapi`, 3

A

Audio (class in twx.botapi), 6
audio (twx.botapi.Message attribute), 4

C

caption (twx.botapi.Video attribute), 8
chat (twx.botapi.Message attribute), 4
Contact (class in twx.botapi), 8
contact (twx.botapi.Message attribute), 5

D

date (twx.botapi.Message attribute), 4
delete_chat_photo (twx.botapi.Message attribute), 5
description (twx.botapi.Error attribute), 12
Document (class in twx.botapi), 6
document (twx.botapi.Message attribute), 4
duration (twx.botapi.Audio attribute), 6
duration (twx.botapi.Video attribute), 7

E

Error (class in twx.botapi), 11
error_code (twx.botapi.Error attribute), 11

F

file_id (twx.botapi.Audio attribute), 6
file_id (twx.botapi.Document attribute), 6
file_id (twx.botapi.PhotoSize attribute), 5
file_id (twx.botapi.Sticker attribute), 7
file_id (twx.botapi.Video attribute), 7
file_info (twx.botapi.InputFile attribute), 9
file_name (twx.botapi.Document attribute), 6
file_size (twx.botapi.Audio attribute), 6
file_size (twx.botapi.Document attribute), 6
file_size (twx.botapi.PhotoSize attribute), 5
file_size (twx.botapi.Sticker attribute), 7
file_size (twx.botapi.Video attribute), 7
first_name (twx.botapi.Contact attribute), 8
first_name (twx.botapi.User attribute), 3
force_reply (twx.botapi.ForceReply attribute), 11
ForceReply (class in twx.botapi), 11

form (twx.botapi.InputFile attribute), 9
forward_date (twx.botapi.Message attribute), 4
forward_from (twx.botapi.Message attribute), 4
forward_message() (in module twx.botapi), 13
from (twx.botapi.Message attribute), 4

G

GET (twx.botapi.RequestMethod attribute), 12
get_me() (in module twx.botapi), 13
get_updates() (in module twx.botapi), 17
get_user_profile_photos() (in module twx.botapi), 17
group_chat_created (twx.botapi.Message attribute), 5
GroupChat (class in twx.botapi), 3

H

height (twx.botapi.PhotoSize attribute), 5
height (twx.botapi.Sticker attribute), 7
height (twx.botapi.Video attribute), 7
hide_keyboard (twx.botapi.ReplyKeyboardHide attribute), 10

I

id (twx.botapi.GroupChat attribute), 3
id (twx.botapi.User attribute), 3
InputFile (class in twx.botapi), 9

K

keyboard (twx.botapi.ReplyKeyboardMarkup attribute), 10

L

last_name (twx.botapi.Contact attribute), 8
last_name (twx.botapi.User attribute), 3
latitude (twx.botapi.Location attribute), 8
left_chat_participant (twx.botapi.Message attribute), 5
Location (class in twx.botapi), 8
location (twx.botapi.Message attribute), 5
longitude (twx.botapi.Location attribute), 8

M

Message (class in twx.botapi), 4

message (twx.botapi.Update attribute), 9
message_id (twx.botapi.Message attribute), 4
mime_type (twx.botapi.Audio attribute), 6
mime_type (twx.botapi.Document attribute), 6
mime_type (twx.botapi.Video attribute), 7

N

new_chat_participant (twx.botapi.Message attribute), 5
new_chat_photo (twx.botapi.Message attribute), 5
new_chat_title (twx.botapi.Message attribute), 5

O

one_time_keyboard (twx.botapi.ReplyKeyboardMarkup attribute), 10

P

phone_number (twx.botapi.Contact attribute), 8
photo (twx.botapi.Message attribute), 4
photos (twx.botapi.UserProfilePhotos attribute), 9
PhotoSize (class in twx.botapi), 5
POST (twx.botapi.RequestMethod attribute), 12

R

reply_to_message (twx.botapi.Message attribute), 4
ReplyKeyboardHide (class in twx.botapi), 10
ReplyKeyboardMarkup (class in twx.botapi), 10
RequestMethod (class in twx.botapi), 12
resize_keyboard (twx.botapi.ReplyKeyboardMarkup attribute), 10

S

selective (twx.botapi.ForceReply attribute), 11
selective (twx.botapi.ReplyKeyboardHide attribute), 11
selective (twx.botapi.ReplyKeyboardMarkup attribute), 10
send_audio() (in module twx.botapi), 14
send_chat_action() (in module twx.botapi), 16
send_document() (in module twx.botapi), 14
send_location() (in module twx.botapi), 16
send_message() (in module twx.botapi), 13
send_photo() (in module twx.botapi), 14
send_sticker() (in module twx.botapi), 15
send_video() (in module twx.botapi), 15
set_webhook() (in module twx.botapi), 17
Sticker (class in twx.botapi), 7
sticker (twx.botapi.Message attribute), 4

T

TelegramBotRPCRequest (class in twx.botapi), 12
text (twx.botapi.Message attribute), 4
thumb (twx.botapi.Document attribute), 6
thumb (twx.botapi.Sticker attribute), 7
thumb (twx.botapi.Video attribute), 7

title (twx.botapi.GroupChat attribute), 3
total_count (twx.botapi.UserProfilePhotos attribute), 9
twx.botapi (module), 3

U

Update (class in twx.botapi), 8
update_id (twx.botapi.Update attribute), 8
User (class in twx.botapi), 3
user_id (twx.botapi.Contact attribute), 8
username (twx.botapi.User attribute), 3
UserProfilePhotos (class in twx.botapi), 9

V

Video (class in twx.botapi), 7
video (twx.botapi.Message attribute), 4

W

width (twx.botapi.PhotoSize attribute), 5
width (twx.botapi.Sticker attribute), 7
width (twx.botapi.Video attribute), 7