
Botlegram

Release 0.0.1

Oct 04, 2020

Contents:

1	Module bottelegram	3
2	Module send	5
3	Exception	9
4	Release History	11
5	Indices and tables	13
	Python Module Index	15
	Index	17

- Release v0.0.2
- Based on [Telegram Bot API](#).
- Source code: [laplacetw/bottelegram](#)
- Stay flexible and keep function naming consistent with the API.

It's good to use.

- Quick Start

```
import bottelegram

BOT_TOKEN = "{{your_bot_token}}"
WEBHOOK = "{{your_host}}" + BOT_TOKEN + '/'
bot = bottelegram.Bot(BOT_TOKEN, WEBHOOK)
>>> * [Bot] Webhook is already set
```

Now we have a Telegram bot

Note: If you'd like to make sure that the Webhook request comes from Telegram, we recommend using a secret path in the URL, e.g. <https://www.example.com/<token>>. Since nobody else knows your bot's token, you can be pretty sure it's from Telegram.

Ref. <https://core.telegram.org/bots/api#setwebhook>

CHAPTER 1

Module bottelegram

All of Bottelegram' functionality can be accessed by the module.

class bottelegram.bot.**Bot** (*token, webhook*)

The class will execute setting webhook when initialization occurs.

Parameters:

1. *webhook* (str): host webhook
2. *token* (str): token of telegram bot

Inherits from:

1. class Send

get_webhook_info ()

For checking webhook setting.

Return: WebhookInfo object

set_webhook ()

For webhook setting.

Return: True

Exception: WebhookError

console_log (*log*)

Print any message to logs immediately. e.g. Heroku logs.

Parameters:

1. *log* (str)

msg_type (*rq_content*)

For checking message types.

Parameters:

1. *rq_content*(json object)

Return: message type(str) or False

CHAPTER 2

Module send

class `bottelegram.utils.send.Send`

A parent class which inherited by class Bot about methods of sending in Telegram Bot API.

send_message (*chat_id, text, parse_mode=None, disable_web_page_preview=False, disable_notification=False, reply_to_message_id=None, reply_markup=None*)

Use this method to send text messages.

Parameters:

1. **chat_id (int/str):** Unique identifier for the target chat or username of the target channel (in the format @channelusername).
2. **text (str):** Text of the message to be sent, 1-4096 characters after entities parsing.
3. **parse_mode (str):** optional. Mode for parsing entities in the message text. See formatting options for more details.
4. **disable_web_page_preview (bool):** optional. Disables link previews for links in this message.
5. **disable_notification (bool):** optional. Sends the message silently. Users will receive a notification with no sound.
6. **reply_to_message_id (int):** optional. If the message is a reply, ID of the original message.
7. **reply_markup (InlineKeyboardMarkup or ReplyKeyboardMarkup or ReplyKeyboardRemove or ForceReply):** optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Return: Message object

forward_message (*chat_id, from_chat_id, message_id, disable_notification=False*)

Use this method to forward messages of any kind.

Parameters:

1. **chat_id (int/str):** Unique identifier for the target chat or username of the target channel (in the format @channelusername).

2. **from_chat_id (int/str):** Unique identifier for the chat where the original message was sent (or channel username in the format @channelusername).
3. **message_id (int):** Message identifier in the chat specified in from_chat_id.
4. **disable_notification (bool):** optional. Sends the message silently. Users will receive a notification with no sound.

Return: Message object

send_photo (*chat_id, photo, caption=None, parse_mode=None, disable_notification=False, reply_to_message_id=None, reply_markup=None*)

Use this method to send photos.

Parameters:

1. **chat_id (int/str):** Unique identifier for the target chat or username of the target channel (in the format @channelusername).
2. **photo (InputFile/str):** Photo to send. Pass a file_id as String to send a photo that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get a photo from the Internet, or upload a new photo using multipart/form-data.
3. **caption (str):** optional. Photo caption (may also be used when resending photos by file_id), 0-1024 characters after entities parsing.
4. **parse_mode (str):** optional. Mode for parsing entities in the message text. See formatting options for more details.
5. **disable_notification (bool):** optional. Sends the message silently. Users will receive a notification with no sound.
6. **reply_to_message_id (int):** optional. If the message is a reply, ID of the original message.
7. **reply_markup (InlineKeyboardMarkup or ReplyKeyboardMarkup or ReplyKeyboardRemove or ForceReply):** optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Return: Message object

Note: To prevent Telegram url cache, we can put timestamp into the url.

e.g. <photo_url>?timestamp=xxxxxxxxxxx

Ref. [stackoverflow](#)

send_audio (*chat_id, audio, caption=None, parse_mode=None, duration=None, performer=None, title=None, thumb=None, disable_notification=False, reply_to_message_id=None, reply_markup=None*)

Use this method to send audio files, if you want Telegram clients to display them in the music player. Your audio must be in the .MP3 or .M4A format. Bots can currently send audio files of up to 50 MB in size, this limit may be changed in the future.

For sending voice messages, use the send_voice() instead.

Parameters:

1. **chat_id (int/str):** Unique identifier for the target chat or username of the target channel (in the format @channelusername).
2. **audio (InputFile/str):** Audio file to send. Pass a file_id as String to send an audio file that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get an audio file from the Internet, or upload a new one using multipart/form-data.

3. **caption (str):** optional. Audio caption, 0-1024 characters after entities parsing.
4. **parse_mode (str):** optional. Mode for parsing entities in the message text. See formatting options for more details.
5. **duration (int):** optional. Duration of the audio in seconds.
6. **performer (str):** optional. Performer.
7. **title (str):** optional. Track name.
8. **thumb (InputFile/str):** optional. Thumbnail of the file sent; can be ignored if thumbnail generation for the file is supported server-side. The thumbnail should be in JPEG format and less than 200 kB in size. A thumbnail's width and height should not exceed 320. Ignored if the file is not uploaded using multipart/form-data. Thumbnails can't be reused and can be only uploaded as a new file, so you can pass "attach://<file_attach_name>" if the thumbnail was uploaded using multipart/form-data under <file_attach_name>.
9. **disable_notification (bool):** optional. Sends the message silently. Users will receive a notification with no sound.
10. **reply_to_message_id (int):** optional. If the message is a reply, ID of the original message.
11. **reply_markup (InlineKeyboardMarkup or ReplyKeyboardMarkup or ReplyKeyboardRemove or ForceReply):** optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Return: Message object

send_voice (*chat_id*, *voice*, *caption=None*, *parse_mode=None*, *duration=None*, *disable_notification=False*, *reply_to_message_id=None*, *reply_markup=None*)

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 Audio or Document). On success, the sent Message is returned. Bots can currently send voice messages of up to 50 MB in size, this limit may be changed in the future.

1. **chat_id (int/str):** Unique identifier for the target chat or username of the target channel (in the format @channelusername).
2. **voice (InputFile/str):** Audio file to send. Pass a file_id as String to send a file that exists on the Telegram servers (recommended), pass an HTTP URL as a String for Telegram to get a file from the Internet, or upload a new one using multipart/form-data.
3. **caption (str):** optional. Audio caption, 0-1024 characters after entities parsing.
4. **parse_mode (str):** optional. Mode for parsing entities in the message text. See formatting options for more details.
5. **duration (int):** optional. Duration of the audio in seconds.
6. **disable_notification (bool):** optional. Sends the message silently. Users will receive a notification with no sound.
7. **reply_to_message_id (int):** optional. If the message is a reply, ID of the original message.
8. **reply_markup (InlineKeyboardMarkup or ReplyKeyboardMarkup or ReplyKeyboardRemove or ForceReply):** optional. Additional interface options. A JSON-serialized object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Return: Message object

CHAPTER 3

Exception

exception `bottlegram.exception.WebhookError`
Raise when webhook setting failed.

CHAPTER 4

Release History

v0.0.2 (2020-10-04)

- **Update methods of sending** `Send.forward_message()`
`Send.send_audio()`
`Send.send_voice()`

v0.0.1 (2020-09-16)

- **Basic setting function.** e.g. `Bot.set_webhook()`
- **Methods of sending** e.g. `Send.send_message()`

CHAPTER 5

Indices and tables

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

b

`bottelegram.bot`, 3
`bottelegram.exception`, 9
`bottelegram.utils.send`, 5

B

`Bot` (*class in bottelegram.bot*), 3
`bottelegram.bot` (*module*), 3
`bottelegram.exception` (*module*), 9
`bottelegram.utils.send` (*module*), 5

C

`console_log()` (*bottelegram.bot.Bot method*), 3

F

`forward_message()` (*bottelegram.utils.send.Send method*), 5

G

`get_webhook_info()` (*bottelegram.bot.Bot method*), 3

M

`msg_type()` (*bottelegram.bot.Bot method*), 3

S

`Send` (*class in bottelegram.utils.send*), 5
`send_audio()` (*bottelegram.utils.send.Send method*), 6
`send_message()` (*bottelegram.utils.send.Send method*), 5
`send_photo()` (*bottelegram.utils.send.Send method*), 6
`send_voice()` (*bottelegram.utils.send.Send method*), 7
`set_webhook()` (*bottelegram.bot.Bot method*), 3

W

`WebhookError`, 9