
Pyrogram

Release 0.13.0

May 31, 2019

1	How the Documentation is Organized	3
2	First Steps	5
3	API Reference	7
4	Meta	9
4.1	Quick Start	9
4.2	Install Guide	10
4.3	Project Setup	11
4.4	Authorization	12
4.5	Calling Methods	13
4.6	Handling Updates	15
4.7	Error Handling	16
4.8	Pyrogram Client	18
4.9	Available Methods	20
4.10	Available Types	55
4.11	Bound Methods	74
4.12	Update Handlers	98
4.13	Decorators	101
4.14	Update Filters	103
4.15	RPC Errors	107
4.16	Using Filters	111
4.17	More on Updates	113
4.18	Configuration File	117
4.19	Smart Plugins	118
4.20	Auto Authorization	124
4.21	Session Settings	125
4.22	Fast Crypto	126
4.23	Text Formatting	127
4.24	Object Serialization	128
4.25	SOCKS5 Proxy	129
4.26	Bots Interaction	130
4.27	MTPROTO vs. Bot API	130
4.28	Test Servers	134
4.29	Advanced Usage	134
4.30	Voice Calls	136

4.31	Pyrogram FAQ	137
4.32	Pyrogram Glossary	141
4.33	Release Notes	141
4.34	Powered by Pyrogram	142
4.35	Support Pyrogram	143
4.36	Functions	143
4.37	Types	232
Python Module Index		461
Index		463

```
from pyrogram import Client, Filters

app = Client("my_account")

@app.on_message(Filters.private)
def hello(client, message):
    message.reply("Hello {}".format(message.from_user.first_name))

app.run()
```

Pyrogram is an elegant, easy-to-use [Telegram](#) client library and framework written from the ground up in Python and C. It enables you to easily create custom apps for both user and bot identities (bot API alternative) via the [MTProto API](#).

CHAPTER 1

How the Documentation is Organized

Contents are organized into self-contained topics and can be all accessed from the sidebar, or by following them in order using the Next button at the end of each page. Here below you can, instead, find a list of the most relevant pages for a quick access.

CHAPTER 2

First Steps

- **Quick Start:** Overview to get you started quickly.
- **Calling Methods:** How to call Pyrogram's methods.
- **Handling Updates:** How to handle Telegram updates.
- **Error Handling:** How to handle API errors correctly.

CHAPTER 3

API Reference

- **Pyrogram Client:** Reference details about the Client class.
- **Available Methods:** List of available high-level methods.
- **Available Types:** List of available high-level types.
- **Bound Methods:** List of convenient bound methods.

- [Pyrogram FAQ](#): Answers to common Pyrogram questions.
- [Pyrogram Glossary](#): List of words with brief explanations.
- [Release Notes](#): Release notes for Pyrogram releases.
- [Powered by Pyrogram](#): Collection of Pyrogram Projects.
- [Support Pyrogram](#): Ways to show your appreciation.

4.1 Quick Start

The next few steps serve as a quick start for all new Pyrogrammers that want to get something done as fast as possible. Let's go!

4.1.1 Get Pyrogram Real Fast

1. Install Pyrogram with `pip3 install -U pyrogram`.
2. Get your own Telegram API key from <https://my.telegram.org/apps>.
3. Open your best text editor and paste the following:

```
from pyrogram import Client

api_id = 12345
api_hash = "0123456789abcdef0123456789abcdef"

with Client("my_account", api_id, api_hash) as app:
    app.send_message("me", "Greetings from **Pyrogram**!")
```

4. Replace `api_id` and `api_hash` values with your own.
5. Save the file as `pyro.py`.
6. Run the script with `python3 pyro.py`

7. Follow the instructions on your terminal to login.
8. Watch Pyrogram send a message to yourself.
9. Join our [community](#).
10. Say, “hi!”.

4.1.2 Enjoy the API

That was just a quick overview that barely scratched the surface! In the next few pages of the introduction, we’ll take a much more in-depth look of what we have just done above.

Feeling eager to continue? You can take a shortcut to [Calling Methods](#) and come back later to learn some more details.

4.2 Install Guide

Being a Python library, **Pyrogram** requires Python to be installed in your system. We recommend using the latest versions of both Python 3 and pip.

- Get **Python 3** from <https://www.python.org/downloads/> (or with your package manager)
- Get **pip** by following the instructions at <https://pip.pypa.io/en/latest/installing/>.

Important: Pyrogram supports **Python 3** only, starting from version 3.4. **PyPy** is supported too.

4.2.1 Install Pyrogram

- The easiest way to install and upgrade Pyrogram to its latest stable version is by using **pip**:

```
$ pip3 install -U pyrogram
```

- or, with **TgCrypto** as extra requirement (recommended):

```
$ pip3 install -U pyrogram[fast]
```

4.2.2 Bleeding Edge

Pyrogram is always evolving, although new releases on PyPI are published only when enough changes are added, but this doesn’t mean you can’t try new features right now!

In case you’d like to try out the latest Pyrogram features, the [GitHub repo](#) is always kept updated with new changes; you can install the development version straight from the `develop` branch using this command (note “develop.zip” in the link):

```
$ pip3 install -U https://github.com/pyrogram/pyrogram/archive/develop.zip
```

4.2.3 Asynchronous

Pyrogram heavily depends on IO-bound network code (it's a cloud-based messaging framework after all), and here's where asyncio shines the most by providing extra performance and efficiency while running on a single OS-level thread only.

A fully asynchronous variant of Pyrogram is therefore available (Python 3.5.3+ required). Use this command to install (note "asyncio.zip" in the link):

```
$ pip3 install -U https://github.com/pyrogram/pyrogram/archive/asyncio.zip
```

Pyrogram's API remains the same and features are kept up to date from the non-async, default develop branch, but you are obviously required Python asyncio knowledge in order to take full advantage of it.

Tip: The idea to turn Pyrogram fully asynchronous is still under consideration, but is wise to expect that in future this would be the one and only way to work with Pyrogram.

You can start using Pyrogram Async variant right now as an excuse to learn more about asynchronous programming and do experiments with it!

Subscribe to [@Pyrogram](#) for news and announcements

4.2.4 Verifying

To verify that Pyrogram is correctly installed, open a Python shell and import it. If no error shows up you are good to go.

```
>>> import pyrogram
>>> pyrogram.__version__
'0.13.0'
```

4.3 Project Setup

We have just [installed Pyrogram](#). In this page we'll discuss what you need to do in order to set up a project with the library. Let's see how it's done.

4.3.1 API Keys

The very first step requires you to obtain a valid Telegram API key (API id/hash pair):

1. Visit <https://my.telegram.org/apps> and log in with your Telegram Account.
2. Fill out the form to register a new Telegram application.
3. Done! The API key consists of two parts: **api_id** and **api_hash**.

Important: The API key is personal and must be kept secret.

Note: The API key is unique for each user, but defines a token for a Telegram *application* you are going to build. This means that you are able to authorize multiple users (and bots too) to access the Telegram database through the MTPROTO API by a single API key.

4.3.2 Configuration

Having the API key from the *previous step* in handy, we can now begin to configure a Pyrogram project. There are two ways to do so, and you can choose what fits better for you:

- First option (recommended): create a new `config.ini` file at the root of your working directory, copy-paste the following and replace the **api_id** and **api_hash** values with your own. This is the preferred method because allows you to keep your credentials out of your code without having to deal with how to load them:

```
[pyrogram]
api_id = 12345
api_hash = 0123456789abcdef0123456789abcdef
```

- Alternatively, you can pass your API key to Pyrogram by simply using the `api_id` and `api_hash` parameters of the `Client` class. This way you can have full control on how to store and load your credentials (e.g., you can load the credentials from the environment variables and directly pass the values into Pyrogram):

```
from pyrogram import Client

app = Client(
    "my_account",
    api_id=12345,
    api_hash="0123456789abcdef0123456789abcdef"
)
```

Note: To keep code snippets clean and concise, from now on it is assumed you are making use of the `config.ini` file, thus, the `api_id` and `api_hash` parameters usage won't be shown anymore.

4.4 Authorization

Once a *project is set up*, you will still have to follow a few steps before you can actually use Pyrogram to make API calls. This section provides all the information you need in order to authorize yourself as user or bot.

4.4.1 User Authorization

In order to use the API, Telegram requires that users be authorized via their phone numbers. Pyrogram automatically manages this process, all you need to do is create an instance of the `Client` class by passing to it a `session_name` of your choice (e.g.: “my_account”) and call the `run()` method:

```
from pyrogram import Client

app = Client("my_account")
app.run()
```


This starts an interactive shell asking you to input your **phone number** (including your **Country Code**) and the **phone code** you will receive in your devices that are already authorized or via SMS:

```
Enter phone number: +39*****
Is "+39*****" correct? (y/n): y
Enter phone code: 32768
Logged in successfully as Dan
```

After successfully authorizing yourself, a new file called `my_account.session` will be created allowing Pyrogram to execute API calls with your identity. This file will be loaded again when you restart your app, and as long as you keep the session alive, Pyrogram won't ask you again to enter your phone number.

Important: Your `*.session` files are personal and must be kept secret.

Note: The code above does nothing except asking for credentials and keeping the client online, hit CTRL+C now to stop your application and keep reading.

4.4.2 Bot Authorization

Bots are a special kind of users that are authorized via their tokens (instead of phone numbers), which are created by the **Bot Father**. Bot tokens replace the users' phone numbers only — you still need to **configure a Telegram API key** with Pyrogram, even when using bots.

The authorization process is automatically managed. All you need to do is choose a `session_name` (can be anything, usually your bot username) and pass your bot token using the `bot_token` parameter. The session file will be named after the session name, which will be `my_bot.session` for the example below.

```
from pyrogram import Client

app = Client(
    "my_bot",
    bot_token="123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11"
)

app.run()
```

4.5 Calling Methods

At this point, we have successfully **installed Pyrogram** and **authorized** our account; we are now aiming towards the core of the library. It's time to start playing with the API!

4.5.1 Basic Usage

Making API method calls with Pyrogram is very simple. Here's an example we are going to examine:

```
from pyrogram import Client

app = Client("my_account")
```

(continues on next page)

(continued from previous page)

```
app.start()

print(app.get_me())
app.send_message("me", "Hi, it's me!")
app.send_location("me", 51.500729, -0.124583)
app.send_sticker("me", "CAADBAADyg4AAvLQYAEYD4F7vcZ43AI")

app.stop()
```

Let's begin by importing the Client class from the Pyrogram package:

```
from pyrogram import Client
```

Now instantiate a new Client object, “my_account” is a session name of your choice:

```
app = Client("my_account")
```

To actually make use of any method, the client has to be started first:

```
app.start()
```

Now, you can call any method you like:

```
print(app.get_me()) # Print information about yourself

# Send messages to yourself:
app.send_message("me", "Hi!") # Text message
app.send_location("me", 51.500729, -0.124583) # Location
app.send_sticker("me", "CAADBAADyg4AAvLQYAEYD4F7vcZ43AI") # Sticker
```

Finally, when done, simply stop the client:

```
app.stop()
```

4.5.2 Context Manager

You can also use Pyrogram's Client in a context manager with the `with` statement. The client will automatically `start()` and `stop()` gracefully, even in case of unhandled exceptions in your code. The example above can be therefore rewritten in a much nicer way:

```
from pyrogram import Client

app = Client("my_account")

with app:
    print(app.get_me())
    app.send_message("me", "Hi there! I'm using **Pyrogram**")
    app.send_location("me", 51.500729, -0.124583)
    app.send_sticker("me", "CAADBAADyg4AAvLQYAEYD4F7vcZ43AI")
```

More examples can be found on [GitHub](#).

4.6 Handling Updates

Calling [API methods](#) sequentially is cool, but how to react when, for example, a new message arrives? This page deals with updates and how to handle such events in Pyrogram. Let's have a look at how they work.

4.6.1 Defining Updates

First, let's define what are these updates. As hinted already, updates are simply events that happen in your Telegram account (incoming messages, new members join, bot button presses, etc...), which are meant to notify you about a new specific state that has changed. These updates are handled by registering one or more callback functions in your app using [Handlers](#).

Each handler deals with a specific event and once a matching update arrives from Telegram, your registered callback function will be called back by the framework and its body executed.

4.6.2 Registering a Handler

To explain how handlers work let's have a look at the most used one, the [MessageHandler](#), which will be in charge for handling [Message](#) updates coming from all around your chats. Every other handler shares the same setup logic; you should not have troubles settings them up once you learn from this section.

4.6.3 Using add_handler()

The `add_handler()` method takes any handler instance that wraps around your defined callback function and registers it in your Client. Here's a full example that prints out the content of a message as soon as it arrives:

```
from pyrogram import Client, MessageHandler

def my_function(client, message):
    print(message)

app = Client("my_account")

my_handler = MessageHandler(my_function)
app.add_handler(my_handler)

app.run()
```

Let's examine these four new pieces. First one: a callback function we defined which accepts two arguments - (*client*, *message*). This will be the function that gets executed every time a new message arrives and Pyrogram will call that function by passing the client instance and the new message instance as argument.

```
def my_function(client, message):
    print(message)
```

Second one: the [MessageHandler](#). This object tells Pyrogram the function we defined above must only handle updates that are in form of a [Message](#):

```
my_handler = MessageHandler(my_function)
```

Third: the method `add_handler()`. This method is used to actually register the handler and let Pyrogram know it needs to be taken into consideration when new updates arrive and the internal dispatching phase begins.

```
app.add_handler(my_handler)
```

Last one, the `run()` method. What this does is simply call `start()` and a special method `idle()` that keeps your main scripts alive until you press CTRL+C; the client will be automatically stopped after that.

```
app.run()
```

4.6.4 Using Decorators

All of the above will become quite verbose, especially in case you have lots of handlers to register. A much nicer way to do so is by decorating your callback function with the `on_message()` decorator.

```
from pyrogram import Client

app = Client("my_account")

@app.on_message()
def my_handler(client, message):
    print(message)

app.run()
```

Note: Due to how these decorators work in Pyrogram, they will wrap your defined callback function in a tuple consisting of (handler, group); this will be the value held by your function identifier (e.g.: `my_function` from the example above).

In case, for some reason, you want to get your own function back after it has been decorated, you need to access `my_function[0].callback`, that is, the `callback` field of the `handler` object which is the first element in the tuple, accessed by bracket notation `[0]`.

4.7 Error Handling

Errors are inevitable when working with the API, and they must be correctly handled with `try..except` blocks in order to control the behaviour of your application. Pyrogram errors all live inside the `errors` package:

```
from pyrogram import errors
```

4.7.1 RPCError

The father of all errors is named `RPCError`. This error exists in form of a Python exception which is directly subclass-ed from Python's main `Exception` and is able to catch all Telegram API related errors. This error is raised every time a method call against Telegram's API was unsuccessful.

```
from pyrogram.errors import RPCError
```

Warning: It must be noted that catching this error is bad practice, especially when no feedback is given (i.e. by logging/printing the full error traceback), because it makes it impossible to understand what went wrong.

4.7.2 Error Categories

The `RPCError` packs together all the possible errors Telegram could raise, but to make things tidier, Pyrogram provides categories of errors, which are named after the common HTTP errors and subclass-ed from the `RPCError`:

```
from pyrogram.errors import BadRequest, Forbidden, ...
```

- 303 - SeeOther
- 400 - BadRequest
- 401 - Unauthorized
- 403 - Forbidden
- 406 - NotAcceptable
- 420 - Flood
- 500 - InternalServerError

4.7.3 Single Errors

For a fine-grained control over every single error, Pyrogram does also expose errors that deal each with a specific issue. For example:

```
from pyrogram.errors import FloodWait
```

These errors subclass directly from the category of errors they belong to, which in turn subclass from the father `RPCError`, thus building a class of error hierarchy such as this:

- **RPCError**
 - **BadRequest**
 - * MessageEmpty
 - * UsernameOccupied
 - * ...
 - **InternalServerError**
 - * RpcCallFail
 - * InterDcCallError
 - * ...
 - ...

4.7.4 Unknown Errors

In case Pyrogram does not know anything yet about a specific error, it raises a special 520 - `UnknownError` exception and logs it in the `unknown_errors.txt` file. Users are invited to report these unknown errors.

4.7.5 Errors with Values

Exception objects may also contain some informative values. For example, `FloodWait` holds the amount of seconds you have to wait before you can try again. The value is always stored in the `x` field of the returned exception object:

```
import time
from pyrogram.errors import FloodWait

try:
    ...
except FloodWait as e:
    time.sleep(e.x) # Wait before trying again
```

4.8 Pyrogram Client

This is the `Client` class. It exposes high-level methods for an easy access to the API.

```
from pyrogram import Client

app = Client("my_account")

with app:
    app.send_message("me", "Hi!")
```

`class pyrogram.Client`

Pyrogram Client, the main means for interacting with Telegram.

Parameters

- **session_name** (`str`) – Name to uniquely identify a session of either a User or a Bot, e.g.: “my_account”. This name will be used to save a file to disk that stores details needed for reconnecting without asking again for credentials.
- **api_id** (`int`, *optional*) – The *api_id* part of your Telegram API Key, as integer. E.g.: 12345 This is an alternative way to pass it if you don’t want to use the *config.ini* file.
- **api_hash** (`str`, *optional*) – The *api_hash* part of your Telegram API Key, as string. E.g.: “0123456789abcdef0123456789abcdef”. This is an alternative way to pass it if you don’t want to use the *config.ini* file.
- **app_version** (`str`, *optional*) – Application version. Defaults to “Pyrogram X.Y.Z” This is an alternative way to set it if you don’t want to use the *config.ini* file.
- **device_model** (`str`, *optional*) – Device model. Defaults to *platform.python_implementation()* + “ ” + *platform.python_version()* This is an alternative way to set it if you don’t want to use the *config.ini* file.
- **system_version** (`str`, *optional*) – Operating System version. Defaults to *platform.system()* + “ ” + *platform.release()* This is an alternative way to set it if you don’t want to use the *config.ini* file.
- **lang_code** (`str`, *optional*) – Code of the language used on the client, in ISO 639-1 standard. Defaults to “en”. This is an alternative way to set it if you don’t want to use the *config.ini* file.
- **ipv6** (`bool`, *optional*) – Pass True to connect to Telegram using IPv6. Defaults to False (IPv4).

- **proxy** (dict, *optional*) – Your SOCKS5 Proxy settings as dict, e.g.: `dict(hostname="11.22.33.44", port=1080, username="user", password="pass")`. `username` and `password` can be omitted if your proxy doesn't require authorization. This is an alternative way to setup a proxy if you don't want to use the `config.ini` file.
- **test_mode** (bool, *optional*) – Enable or disable login to the test servers. Defaults to False. Only applicable for new sessions and will be ignored in case previously created sessions are loaded.
- **bot_token** (str, *optional*) – Pass your Bot API token to create a bot session, e.g.: "123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11" Only applicable for new sessions.
- **phone_number** (str | callable, *optional*) – Pass your phone number as string (with your Country Code prefix included) to avoid entering it manually. Or pass a callback function which accepts no arguments and must return the correct phone number as string (e.g., "391234567890"). Only applicable for new sessions.
- **phone_code** (str | callable, *optional*) – Pass the phone code as string (for test numbers only) to avoid entering it manually. Or pass a callback function which accepts a single positional argument (`phone_number`) and must return the correct phone code as string (e.g., "12345"). Only applicable for new sessions.
- **password** (str, *optional*) – Pass your Two-Step Verification password as string (if you have one) to avoid entering it manually. Or pass a callback function which accepts a single positional argument (`password_hint`) and must return the correct password as string (e.g., "password"). Only applicable for new sessions.
- **recovery_code** (callable, *optional*) – Pass a callback function which accepts a single positional argument (`email_pattern`) and must return the correct password recovery code as string (e.g., "987654"). Only applicable for new sessions.
- **force_sms** (str, *optional*) – Pass True to force Telegram sending the authorization code via SMS. Only applicable for new sessions.
- **first_name** (str, *optional*) – Pass a First Name as string to avoid entering it manually. Or pass a callback function which accepts no arguments and must return the correct name as string (e.g., "Dan"). It will be used to automatically create a new Telegram account in case the phone number you passed is not registered yet. Only applicable for new sessions.
- **last_name** (str, *optional*) – Same purpose as `first_name`; pass a Last Name to avoid entering it manually. It can be an empty string: "". Only applicable for new sessions.
- **workers** (int, *optional*) – Thread pool size for handling incoming updates. Defaults to 4.
- **workdir** (str, *optional*) – Define a custom working directory. The working directory is the location in your filesystem where Pyrogram will store your session files. Defaults to "." (current directory).
- **config_file** (str, *optional*) – Path of the configuration file. Defaults to `./config.ini`
- **plugins** (dict, *optional*) – Your Smart Plugins settings as dict, e.g.: `dict(root="plugins")`. This is an alternative way to setup plugins if you don't want to use the `config.ini` file.
- **no_updates** (bool, *optional*) – Pass True to completely disable incoming updates for the current session. When updates are disabled your client can't receive any new message. Useful for batch programs that don't need to deal with updates. Defaults to False (updates enabled and always received).

- **takeout** (*bool, optional*) – Pass True to let the client use a takeout session instead of a normal one, implies *no_updates=True*. Useful for exporting your Telegram data. Methods invoked inside a takeout session (such as *get_history*, *download_media*, ...) are less prone to throw FloodWait exceptions. Only available for users, bots will ignore this parameter. Defaults to False (normal session).

4.9 Available Methods

All Pyrogram methods listed here are bound to a *Client* instance.

```
from pyrogram import Client

app = Client("my_account")

with app:
    app.send_message("haskell", "hi")
```

4.9.1 Utilities

<i>start</i>	Start the Client.
<i>stop</i>	Stop the Client.
<i>restart</i>	Restart the Client.
<i>idle</i>	Block the main script execution until a signal (e.g.: from CTRL+C) is received.
<i>run</i>	Start the Client and automatically idle the main script.
<i>add_handler</i>	Register an update handler.
<i>remove_handler</i>	Remove a previously-registered update handler.
<i>send</i>	Send raw Telegram queries.
<i>resolve_peer</i>	Get the InputPeer of a known peer id.
<i>save_file</i>	Upload a file onto Telegram servers, without actually sending the message to anyone.
<i>stop_transmission</i>	Stop downloading or uploading a file.

4.9.2 Messages

<i>send_message</i>	Send text messages.
<i>forward_messages</i>	Forward messages of any kind.
<i>send_photo</i>	Send photos.
<i>send_audio</i>	Send audio files.
<i>send_document</i>	Send generic files.
<i>send_sticker</i>	Send .webp stickers.
<i>send_video</i>	Send video files.
<i>send_animation</i>	Send animation files (animation or H.264/MPEG-4 AVC video without sound).
<i>send_voice</i>	Send audio files.
<i>send_video_note</i>	Send video messages.
<i>send_media_group</i>	Send a group of photos or videos as an album.
<i>send_location</i>	Send points on the map.

Continued on next page

Table 2 – continued from previous page

<code>send_venue</code>	Send information about a venue.
<code>send_contact</code>	Send phone contacts.
<code>send_cached_media</code>	Send any media stored on the Telegram servers using a <code>file_id</code> .
<code>send_chat_action</code>	Tell the other party that something is happening on your side.
<code>edit_message_text</code>	Edit text messages.
<code>edit_message_caption</code>	Edit captions of messages.
<code>edit_message_reply_markup</code>	Edit only the reply markup of messages sent by the bot or via the bot (for inline bots).
<code>edit_message_media</code>	Edit audio, document, photo, or video messages.
<code>delete_messages</code>	Delete messages, including service messages.
<code>get_messages</code>	Get one or more messages that belong to a specific chat.
<code>get_history</code>	Retrieve a chunk of the history of a chat.
<code>get_history_count</code>	Get the total count of messages in a chat.
<code>iter_history</code>	Iterate through a chat history sequentially.
<code>send_poll</code>	Send a new poll.
<code>vote_poll</code>	Vote a poll.
<code>stop_poll</code>	Stop a poll which was sent by you.
<code>retract_vote</code>	Retract your vote in a poll.
<code>download_media</code>	Download the media from a message.

4.9.3 Chats

<code>join_chat</code>	Join a group chat or channel.
<code>leave_chat</code>	Leave a group chat or channel.
<code>kick_chat_member</code>	Kick a user from a group, a supergroup or a channel.
<code>unban_chat_member</code>	Unban a previously kicked user in a supergroup or channel.
<code>restrict_chat_member</code>	Restrict a user in a supergroup.
<code>promote_chat_member</code>	Promote or demote a user in a supergroup or a channel.
<code>export_chat_invite_link</code>	Generate a new invite link for a chat; any previously generated link is revoked.
<code>set_chat_photo</code>	Set a new profile photo for the chat.
<code>delete_chat_photo</code>	Delete a chat photo.
<code>set_chat_title</code>	Change the title of a chat.
<code>set_chat_description</code>	Change the description of a supergroup or a channel.
<code>pin_chat_message</code>	Pin a message in a group, channel or your own chat.
<code>unpin_chat_message</code>	Unpin a message in a group, channel or your own chat.
<code>get_chat</code>	Get up to date information about a chat.
<code>get_chat_member</code>	Get information about one member of a chat.
<code>get_chat_members</code>	Get a chunk of the members list of a chat.
<code>get_chat_members_count</code>	Get the number of members in a chat.
<code>iter_chat_members</code>	Iterate through the members of a chat sequentially.
<code>get_dialogs</code>	Get a chunk of the user's dialogs.
<code>iter_dialogs</code>	Iterate through a user's dialogs sequentially.
<code>get_dialogs_count</code>	Get the total count of your dialogs.
<code>restrict_chat</code>	Restrict a chat.
<code>update_chat_username</code>	Update a channel or a supergroup username.

4.9.4 Users

<code>get_me</code>	Get your own user identity.
<code>get_users</code>	Get information about a user.
<code>get_user_profile_photos</code>	Get a list of profile pictures for a user.
<code>get_user_profile_photos_count</code>	Get the total count of profile pictures for a user.
<code>set_user_profile_photo</code>	Set a new profile photo.
<code>delete_user_profile_photos</code>	Delete your own profile photos.
<code>update_username</code>	Update your own username.

4.9.5 Contacts

<code>add_contacts</code>	Add contacts to your Telegram address book.
<code>get_contacts</code>	Get contacts from your Telegram address book.
<code>get_contacts_count</code>	Get the total count of contacts from your Telegram address book.
<code>delete_contacts</code>	Delete contacts from your Telegram address book.

4.9.6 Password

<code>enable_cloud_password</code>	Enable the Two-Step Verification security feature (Cloud Password) on your account.
<code>change_cloud_password</code>	Change your Two-Step Verification password (Cloud Password) with a new one.
<code>remove_cloud_password</code>	Turn off the Two-Step Verification security feature (Cloud Password) on your account.

4.9.7 Bots

<code>get_inline_bot_results</code>	Get bot results via inline queries.
<code>send_inline_bot_result</code>	Send an inline bot result.
<code>answer_callback_query</code>	Send answers to callback queries sent from inline keyboards.
<code>answer_inline_query</code>	Send answers to an inline query.
<code>request_callback_answer</code>	Request a callback answer from bots.
<code>send_game</code>	Send a game.
<code>set_game_score</code>	Set the score of the specified user in a game.
<code>get_game_high_scores</code>	Get data for high score tables.

`Client.start()`
Start the Client.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ConnectionError` – In case you try to start an already started Client.

`Client.stop()`
Stop the Client.

Raises `ConnectionError` – In case you try to stop an already stopped Client.

`Client.restart()`
Restart the Client.

Raises `ConnectionError` – In case you try to restart a stopped Client.

`Client.idle()`
Block the main script execution until a signal (e.g.: from CTRL+C) is received. Once the signal is received, the client will automatically stop and the main script will continue its execution.

This is used after starting one or more clients and is useful for event-driven applications only, that are, applications which react upon incoming Telegram updates through handlers, rather than executing a set of methods sequentially.

The way Pyrogram works, will keep your handlers in a pool of workers, which are executed concurrently outside the main script; calling `idle()` will ensure the client(s) will be kept alive by not letting the main script to end, until you decide to quit.

Parameters `stop_signals` (`tuple`, *optional*) – Iterable containing signals the signal handler will listen to. Defaults to (SIGINT, SIGTERM, SIGABRT).

`Client.run()`
Start the Client and automatically idle the main script.

This is a convenience method that literally just calls `start()` and `idle()`. It makes running a client less verbose, but is not suitable in case you want to run more than one client in a single main script, since `idle()` will block.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.add_handler()`
Register an update handler.

You can register multiple handlers, but at most one handler within a group will be used for a single update. To handle the same update more than once, register your handler using a different group id (lower group id == higher priority).

Parameters

- **handler** (`Handler`) – The handler to be registered.
- **group** (`int`, *optional*) – The group identifier, defaults to 0.

Returns `tuple` – A tuple consisting of (handler, group).

`Client.remove_handler()`
Remove a previously-registered update handler.

Make sure to provide the right group that the handler was added in. You can use the return value of the `add_handler()` method, a tuple of (handler, group), and pass it directly.

Parameters

- **handler** (`Handler`) – The handler to be removed.
- **group** (`int`, *optional*) – The group identifier, defaults to 0.

`Client.send()`
Send raw Telegram queries.

This method makes it possible to manually call every single Telegram API method in a low-level manner. Available functions are listed in the `functions` package and may accept compound data types from `types` as well as bare types such as `int`, `str`, etc...

Note: This is a utility method intended to be used **only** when working with raw *functions* (i.e: a Telegram API method you wish to use which is not available yet in the Client class as an easy-to-use method).

Parameters

- **data** (RawFunction) – The API Schema function filled with proper arguments.
- **retries** (int) – Number of retries.
- **timeout** (float) – Timeout in seconds.

Returns RawType – The raw type response generated by the query.

Raises RPCError – In case of a Telegram RPC error.

`Client.resolve_peer()`

Get the InputPeer of a known peer id. Useful whenever an InputPeer type is required.

Note: This is a utility method intended to be used **only** when working with raw *functions* (i.e: a Telegram API method you wish to use which is not available yet in the Client class as an easy-to-use method).

Parameters **peer_id** (int | str) – The peer id you want to extract the InputPeer from. Can be a direct id (int), a username (str) or a phone number (str).

Returns InputPeer – On success, the resolved peer id is returned in form of an InputPeer object.

Raises

- RPCError – In case of a Telegram RPC error.
- KeyError – In case the peer doesn't exist in the internal database.

`Client.save_file()`

Upload a file onto Telegram servers, without actually sending the message to anyone. Useful whenever an InputFile type is required.

Note: This is a utility method intended to be used **only** when working with raw *functions* (i.e: a Telegram API method you wish to use which is not available yet in the Client class as an easy-to-use method).

Parameters

- **path** (str) – The path of the file you want to upload that exists on your local machine.
- **file_id** (int, *optional*) – In case a file part expired, pass the file_id and the file_part to retry uploading that specific chunk.
- **file_part** (int, *optional*) – In case a file part expired, pass the file_id and the file_part to retry uploading that specific chunk.
- **progress** (callable, *optional*) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (tuple, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a chat_id and a message_id in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple, optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns *InputFile* – On success, the uploaded file is returned in form of an *InputFile* object.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.stop_transmission()`

Stop downloading or uploading a file. Must be called inside a progress callback function.

`Client.send_message()`

Send text messages.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **text** (*str*) – Text of the message to be sent.
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your message. Defaults to “markdown”.
- **disable_web_page_preview** (*bool, optional*) – Disables link previews for links in this message.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Returns *Message* – On success, the sent text message is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.forward_messages()`

Forward messages of any kind.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **from_chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the source chat where the original message was sent. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).

- **message_ids** (iterable) – A list of Message identifiers in the chat specified in *from_chat_id* or a single message id. Iterators and Generators are also accepted.
- **disable_notification** (bool, optional) – Sends the message silently. Users will receive a notification with no sound.
- **as_copy** (bool, optional) – Pass True to forward messages without the forward header (i.e.: send a copy of the message content). Defaults to False.
- **remove_caption** (bool, optional) – If set to True and *as_copy* is enabled as well, media captions are not preserved when copying the message. Has no effect if *as_copy* is not enabled. Defaults to False.

Returns *Message* | *Messages* – In case *message_ids* was an integer, the single forwarded message is returned, otherwise, in case *message_ids* was an iterable, the returned value will be an object containing a list of messages, even if such iterable contained just a single element.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_photo()`
Send photos.

Parameters

- **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (str).
- **photo** (str) – Photo to send. Pass a file_id as string to send a photo that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get a photo from the Internet, or pass a file path as string to upload a new photo that exists on your local machine.
- **caption** (bool, optional) – Photo caption, 0-1024 characters.
- **parse_mode** (str, optional) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **ttl_seconds** (int, optional) – Self-Destruct Timer. If you set a timer, the photo will self-destruct in *ttl_seconds* seconds after it was viewed.
- **disable_notification** (bool, optional) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (int, optional) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, optional) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (callable, optional) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (tuple, optional) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a chat_id and a message_id in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.

- **current** (int) – The amount of bytes uploaded so far.
- **total** (int) – The size of the file.
- ***args** (tuple, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns *Message* | None – On success, the sent photo message is returned, otherwise, in case the upload is deliberately stopped with *stop_transmission()*, None is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_audio()`
Send audio files.

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

Parameters

- **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (str).
- **audio** (str) – Audio file to send. Pass a file_id as string to send an audio file that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get an audio file from the Internet, or pass a file path as string to upload a new audio file that exists on your local machine.
- **caption** (str, *optional*) – Audio caption, 0-1024 characters.
- **parse_mode** (str, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **duration** (int, *optional*) – Duration of the audio in seconds.
- **performer** (str, *optional*) – Performer.
- **title** (str, *optional*) – Track name.
- **thumb** (str, *optional*) – Thumbnail of the music file album cover. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail’s width and height should not exceed 320 pixels. Thumbnails can’t be reused and can be only uploaded as a new file.
- **disable_notification** (bool, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (int, *optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (callable, *optional*) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (tuple, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a chat_id and a message_id in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (int) – The amount of bytes uploaded so far.
- **total** (int) – The size of the file.
- ***args** (tuple, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns *Message* | None – On success, the sent audio message is returned, otherwise, in case the upload is deliberately stopped with *stop_transmission()*, None is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_document()`

Send generic files.

Parameters

- **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (str).
- **document** (str) – File to send. Pass a file_id as string to send a file that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get a file from the Internet, or pass a file path as string to upload a new file that exists on your local machine.
- **thumb** (str, *optional*) – Thumbnail of the file sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail’s width and height should not exceed 320 pixels. Thumbnails can’t be reused and can be only uploaded as a new file.
- **caption** (str, *optional*) – Document caption, 0-1024 characters.
- **parse_mode** (str, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **disable_notification** (bool, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (int, *optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (callable, *optional*) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (tuple, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a chat_id and a message_id in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (int) – The amount of bytes uploaded so far.
- **total** (int) – The size of the file.

- ***args** (tuple, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns *Message* | None – On success, the sent document message is returned, otherwise, in case the upload is deliberately stopped with *stop_transmission()*, None is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_sticker()`

Send .webp stickers.

Parameters

- **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (str).
- **sticker** (str) – Sticker to send. Pass a file_id as string to send a sticker that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get a .webp sticker file from the Internet, or pass a file path as string to upload a new sticker that exists on your local machine.
- **disable_notification** (bool, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (int, *optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (callable, *optional*) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (tuple, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a chat_id and a message_id in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (int) – The amount of bytes uploaded so far.
- **total** (int) – The size of the file.
- ***args** (tuple, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns *Message* | None – On success, the sent sticker message is returned, otherwise, in case the upload is deliberately stopped with *stop_transmission()*, None is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_video()`

Send video files.

Parameters

- **chat_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **video** (*str*) – Video to send. Pass a *file_id* as string to send a video that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get a video from the Internet, or pass a file path as string to upload a new video that exists on your local machine.
- **caption** (*str*, *optional*) – Video caption, 0-1024 characters.
- **parse_mode** (*str*, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **duration** (*int*, *optional*) – Duration of sent video in seconds.
- **width** (*int*, *optional*) – Video width.
- **height** (*int*, *optional*) – Video height.
- **thumb** (*str*, *optional*) – Thumbnail of the video sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail’s width and height should not exceed 320 pixels. Thumbnails can’t be reused and can be only uploaded as a new file.
- **supports_streaming** (*bool*, *optional*) – Pass True, if the uploaded video is suitable for streaming.
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int*, *optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable*, *optional*) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple*, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple*, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns *Message* | *None* – On success, the sent video message is returned, otherwise, in case the upload is deliberately stopped with *stop_transmission()*, *None* is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_animation()`

Send animation files (animation or H.264/MPEG-4 AVC video without sound).

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **animation** (`str`) – Animation to send. Pass a `file_id` as string to send an animation that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get an animation from the Internet, or pass a file path as string to upload a new animation that exists on your local machine.
- **caption** (`str`, *optional*) – Animation caption, 0-1024 characters.
- **parse_mode** (`str`, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **duration** (`int`, *optional*) – Duration of sent animation in seconds.
- **width** (`int`, *optional*) – Animation width.
- **height** (`int`, *optional*) – Animation height.
- **thumb** (`str`, *optional*) – Thumbnail of the animation file sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail’s width and height should not exceed 320 pixels. Thumbnails can’t be reused and can be only uploaded as a new file.
- **disable_notification** (`bool`, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (`int`, *optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (`InlineKeyboardMarkup` | `ReplyKeyboardMarkup` | `ReplyKeyboardRemove` | `ForceReply`, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (`callable`, *optional*) – Pass a callback function to view the upload progress. The function must take (`client`, `current`, `total`, `*args`) as positional arguments (look at the section below for a detailed description).
- **progress_args** (`tuple`, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a `chat_id` and a `message_id` in order to edit a message with the updated progress.

Other Parameters

- **client** (`Client`) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (`int`) – The amount of bytes uploaded so far.
- **total** (`int`) – The size of the file.
- ***args** (`tuple`, *optional*) – Extra custom arguments as defined in the `progress_args` parameter. You can either keep `*args` or add every single extra argument in your function signature.

Returns `Message` | `None` – On success, the sent animation message is returned, otherwise, in case the upload is deliberately stopped with `stop_transmission()`, `None` is returned.

Raises `RPCError` – In case of a Telegram RPC error.

```
Client.send_voice()
    Send audio files.
```

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **voice** (`str`) – Audio file to send. Pass a `file_id` as string to send an audio that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get an audio from the Internet, or pass a file path as string to upload a new audio that exists on your local machine.
- **caption** (`str`, *optional*) – Voice message caption, 0-1024 characters.
- **parse_mode** (`str`, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **duration** (`int`, *optional*) – Duration of the voice message in seconds.
- **disable_notification** (`bool`, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (`int`, *optional*) – If the message is a reply, ID of the original message
- **reply_markup** (`InlineKeyboardMarkup` | `ReplyKeyboardMarkup` | `ReplyKeyboardRemove` | `ForceReply`, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (`callable`, *optional*) – Pass a callback function to view the upload progress. The function must take (`client`, `current`, `total`, `*args`) as positional arguments (look at the section below for a detailed description).
- **progress_args** (`tuple`, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a `chat_id` and a `message_id` in order to edit a message with the updated progress.

Other Parameters

- **client** (`Client`) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (`int`) – The amount of bytes uploaded so far.
- **total** (`int`) – The size of the file.
- ***args** (`tuple`, *optional*) – Extra custom arguments as defined in the `progress_args` parameter. You can either keep `*args` or add every single extra argument in your function signature.

Returns `Message` | `None` – On success, the sent voice message is returned, otherwise, in case the upload is deliberately stopped with `stop_transmission()`, `None` is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.send_video_note()`
Send video messages.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).

- **video_note** (*str*) – Video note to send. Pass a *file_id* as string to send a video note that exists on the Telegram servers, or pass a file path as string to upload a new video note that exists on your local machine. Sending video notes by a URL is currently unsupported.
- **duration** (*int*, *optional*) – Duration of sent video in seconds.
- **length** (*int*, *optional*) – Video width and height.
- **thumb** (*str*, *optional*) – Thumbnail of the video sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail's width and height should not exceed 320 pixels. Thumbnails can't be reused and can be only uploaded as a new file.
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int*, *optional*) – If the message is a reply, ID of the original message
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable*, *optional*) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple*, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple*, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns *Message* | *None* – On success, the sent video note message is returned, otherwise, in case the upload is deliberately stopped with *stop_transmission()*, *None* is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_media_group()`

Send a group of photos or videos as an album.

Parameters

- **chat_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **media** (List of *InputMediaPhoto* and *InputMediaVideo*) – A list describing photos and videos to be sent, must include 2–10 items.
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.

- **reply_to_message_id** (*int*, *optional*) – If the message is a reply, ID of the original message.

Returns *Messages* – On success, an object is returned containing all the single messages sent.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_location()`

Send points on the map.

Parameters

- **chat_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **latitude** (*float*) – Latitude of the location.
- **longitude** (*float*) – Longitude of the location.
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int*, *optional*) – If the message is a reply, ID of the original message
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Returns *Message* – On success, the sent location message is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_venue()`

Send information about a venue.

Parameters

- **chat_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **latitude** (*float*) – Latitude of the venue.
- **longitude** (*float*) – Longitude of the venue.
- **title** (*str*) – Name of the venue.
- **address** (*str*) – Address of the venue.
- **foursquare_id** (*str*, *optional*) – Foursquare identifier of the venue.
- **foursquare_type** (*str*, *optional*) – Foursquare type of the venue, if known. (For example, “arts_entertainment/default”, “arts_entertainment/aquarium” or “food/icecream”.)
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int*, *optional*) – If the message is a reply, ID of the original message

- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Returns *Message* – On success, the sent venue message is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_contact()`

Send phone contacts.

Parameters

- **chat_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **phone_number** (*str*) – Contact’s phone number.
- **first_name** (*str*) – Contact’s first name.
- **last_name** (*str*, *optional*) – Contact’s last name.
- **vcard** (*str*, *optional*) – Additional data about the contact in the form of a vCard, 0-2048 bytes
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int*, *optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Returns *Message* – On success, the sent contact message is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_cached_media()`

Send any media stored on the Telegram servers using a *file_id*.

This convenience method works with any valid *file_id* only. It does the same as calling the relevant method for sending media using a *file_id*, thus saving you from the hassle of using the correct method for the media the *file_id* is pointing to.

Parameters

- **chat_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **file_id** (*str*) – Media to send. Pass a *file_id* as string to send a media that exists on the Telegram servers.
- **caption** (*bool*, *optional*) – Media caption, 0-1024 characters.
- **parse_mode** (*str*, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.

- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Returns *Message* – On success, the sent media message is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_chat_action()`

Tell the other party that something is happening on your side.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **action** (*str*) – 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, “*record_video_note*” or “*upload_video_note*” for video notes, “*choose_contact*” for contacts, “*playing*” for games or “*cancel*” to cancel any chat action currently displayed.

Returns *bool* – On success, True is returned.

Raises

- *RPCError* – In case of a Telegram RPC error.
- *ValueError* – In case the provided string is not a valid ChatAction.

`Client.edit_message_text()`

Edit text messages.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **message_id** (*int*) – Message identifier in the chat specified in *chat_id*.
- **text** (*str*) – New text of the message.
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your message. Defaults to “markdown”.
- **disable_web_page_preview** (*bool, optional*) – Disables link previews for links in this message.
- **reply_markup** (*InlineKeyboardMarkup, optional*) – An *InlineKeyboardMarkup* object.

Returns *Message* – On success, the edited message is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.edit_message_caption()`

Edit captions of messages.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **message_id** (`int`) – Message identifier in the chat specified in `chat_id`.
- **caption** (`str`) – New caption of the message.
- **parse_mode** (`str`, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **reply_markup** (*`InlineKeyboardMarkup`*, *optional*) – An `InlineKeyboardMarkup` object.

Returns `Message` – On success, the edited message is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.edit_message_reply_markup()`

Edit only the reply markup of messages sent by the bot or via the bot (for inline bots).

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **message_id** (`int`) – Message identifier in the chat specified in `chat_id`.
- **reply_markup** (*`InlineKeyboardMarkup`*, *optional*) – An `InlineKeyboardMarkup` object.

Returns `Message` | `bool` – In case the edited message is sent by the bot, the edited message is returned, otherwise, `True` is returned in case the edited message is sent by the user.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.edit_message_media()`

Edit audio, document, photo, or video messages.

If a message is a part of a message album, then it can be edited only to a photo or a video. Otherwise, message type can be changed arbitrarily. When inline message is edited, new file can't be uploaded. Use previously uploaded file via its `file_id` or specify a URL. On success, if the edited message was sent by the bot, the edited `Message` is returned, otherwise `True` is returned.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **message_id** (`int`) – Message identifier in the chat specified in `chat_id`.
- **media** (*`InputMedia`*) – One of the `InputMedia` objects describing an animation, audio, document, photo or video.
- **reply_markup** (*`InlineKeyboardMarkup`*, *optional*) – An `InlineKeyboardMarkup` object.

Returns `Message` – On success, the edited message is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.delete_messages()`

Delete messages, including service messages.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **message_ids** (`iterable`) – A list of Message identifiers to delete or a single message id. Iterators and Generators are also accepted.
- **revoke** (`bool`, *optional*) – Deletes messages on both parts. This is only for private cloud chats and normal groups, messages on channels and supergroups are always revoked (i.e.: deleted for everyone). Defaults to `True`.

Returns `bool` – `True` on success, `False` otherwise.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.get_messages()`

Get one or more messages that belong to a specific chat. You can retrieve up to 200 messages at once.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **message_ids** (`iterable`, *optional*) – Pass a single message identifier or a list of message ids (as integers) to get the content of the message themselves. Iterators and Generators are also accepted.
- **reply_to_message_ids** (`iterable`, *optional*) – Pass a single message identifier or a list of message ids (as integers) to get the content of the previous message you replied to using this message. Iterators and Generators are also accepted. If *message_ids* is set, this argument will be ignored.
- **replies** (`int`, *optional*) – The number of subsequent replies to get for each message. Pass 0 for no reply at all or -1 for unlimited replies. Defaults to 1.

Returns `Message` | `Messages` – In case *message_ids* was an integer, the single requested message is returned, otherwise, in case *message_ids* was an iterable, the returned value will be an object containing a list of messages, even if such iterable contained just a single element.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.get_history()`

Retrieve a chunk of the history of a chat.

You can get up to 100 messages at once. For a more convenient way of getting a chat history see `iter_history()`.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **limit** (`int`, *optional*) – Limits the number of messages to be retrieved. By default, the first 100 messages are returned.

- **offset** (*int, optional*) – Sequential number of the first message to be returned. Defaults to 0 (most recent message). Negative values are also accepted and become useful in case you set `offset_id` or `offset_date`.
- **offset_id** (*int, optional*) – Pass a message identifier as offset to retrieve only older messages starting from that message.
- **offset_date** (*int, optional*) – Pass a date in Unix time as offset to retrieve only older messages starting from that date.
- **reverse** (*bool, optional*) – Pass True to retrieve the messages in reversed order (from older to most recent).

Returns *Messages* – On success, an object containing a list of the retrieved messages.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.get_history_count()`

Get the total count of messages in a chat.

Note: Due to Telegram latest internal changes, the server can't reliably find anymore the total count of messages a **private** or a **basic group** chat has with a single method call. To overcome this limitation, Pyrogram has to iterate over all the messages. Channels and supergroups are not affected by this limitation.

Parameters `chat_id` (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat.

Returns *int* – On success, the chat history count is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.iter_history()`

Iterate through a chat history sequentially.

This convenience method does the same as repeatedly calling `get_history()` in a loop, thus saving you from the hassle of setting up boilerplate code. It is useful for getting the whole chat history with a single call.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **limit** (*int, optional*) – Limits the number of messages to be retrieved. By default, no limit is applied and all messages are returned.
- **offset** (*int, optional*) – Sequential number of the first message to be returned.. Negative values are also accepted and become useful in case you set `offset_id` or `offset_date`.
- **offset_id** (*int, optional*) – Identifier of the first message to be returned.
- **offset_date** (*int, optional*) – Pass a date in Unix time as offset to retrieve only older messages starting from that date.
- **reverse** (*bool, optional*) – Pass True to retrieve the messages in reversed order (from older to most recent).

Returns *Generator* – A generator yielding *Message* objects.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.send_poll()`

Send a new poll.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **question** (`str`) – Poll question, 1-255 characters.
- **options** (List of `str`) – List of answer options, 2-10 strings 1-100 characters each.
- **disable_notification** (`bool`, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (`int`, *optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (`InlineKeyboardMarkup` | `ReplyKeyboardMarkup` | `ReplyKeyboardRemove` | `ForceReply`, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Returns `Message` – On success, the sent poll message is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.vote_poll()`

Vote a poll.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **message_id** (`int`) – Identifier of the original message with the poll.
- **option** (`int`) – Index of the poll option you want to vote for (0 to 9).

Returns `Poll` – On success, the poll with the chosen option is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.stop_poll()`

Stop a poll which was sent by you.

Stopped polls can't be reopened and nobody will be able to vote in it anymore.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **message_id** (`int`) – Identifier of the original message with the poll.
- **reply_markup** (`InlineKeyboardMarkup`, *optional*) – An `InlineKeyboardMarkup` object.

Returns `Poll` – On success, the stopped poll with the final results is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.retract_vote()`

Retract your vote in a poll.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **message_id** (`int`) – Identifier of the original message with the poll.

Returns `Poll` – On success, the poll with the retracted vote is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.download_media()`

Download the media from a message.

Parameters

- **message** (`Message` | `str`) – Pass a `Message` containing the media, the media itself (`message.audio`, `message.video`, ...) or the file id as string.
- **file_name** (`str`, *optional*) – A custom `file_name` to be used instead of the one provided by Telegram. By default, all files are downloaded in the `downloads` folder in your working directory. You can also specify a path for downloading files in a custom location: paths that end with “/” are considered directories. All non-existent folders will be created automatically.
- **block** (`bool`, *optional*) – Blocks the code execution until the file has been downloaded. Defaults to `True`.
- **progress** (*callable*) – Pass a callback function to view the download progress. The function must take (`client`, `current`, `total`, `*args`) as positional arguments (look at the section below for a detailed description).
- **progress_args** (`tuple`) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a `chat_id` and a `message_id` in order to edit a message with the updated progress.

Other Parameters

- **client** (`Client`) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (`int`) – The amount of bytes downloaded so far.
- **total** (`int`) – The size of the file.
- ***args** (`tuple`, *optional*) – Extra custom arguments as defined in the `progress_args` parameter. You can either keep `*args` or add every single extra argument in your function signature.

Returns `str` | `None` – On success, the absolute path of the downloaded file is returned, otherwise, in case the download failed or was deliberately stopped with `stop_transmission()`, `None` is returned.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` if the message doesn’t contain any downloadable media

`Client.join_chat()`

Join a group chat or channel.

Parameters **chat_id** (`str`) – Unique identifier for the target chat in form of a `t.me/joinchat/` link or username of the target channel/supergroup (in the format `@username`).

Returns `Chat` – On success, a chat object is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.leave_chat()`

Leave a group chat or channel.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier for the target chat or username of the target channel/supergroup (in the format `@username`).
- **delete** (`bool`, *optional*) – Deletes the group chat dialog after leaving (for simple group chats, not supergroups).

Raises `RPCError` – In case of a Telegram RPC error.

`Client.kick_chat_member()`

Kick a user from a group, a supergroup or a channel. In the case of supergroups and channels, the user will not be able to return to the group on their own using invite links, etc., unless unbanned first. You must be an administrator in the chat for this to work and must have the appropriate admin rights.

Note: In regular groups (non-supergroups), this method will only work if the “All Members Are Admins” setting is off in the target group. Otherwise members may only be removed by the group’s creator or by the member that added them.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat.
- **user_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target user. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **until_date** (`int`, *optional*) – Date when the user will be unbanned, unix time. If user is banned for more than 366 days or less than 30 seconds from the current time they are considered to be banned forever. Defaults to 0 (ban forever).

Returns `Message` | `bool` – On success, a service message will be returned (when applicable), otherwise, in case a message object couldn’t be returned, `True` is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.unban_chat_member()`

Unban a previously kicked user in a supergroup or channel. The user will **not** return to the group or channel automatically, but will be able to join via link, etc. You must be an administrator for this to work.

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat.
- **user_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target user. For a contact that exists in your Telegram address book you can use his phone number (`str`).

Returns `bool` – `True` on success.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.restrict_chat_member()`

Restrict a user in a supergroup.

The bot must be an administrator in the supergroup for this to work and must have the appropriate admin rights. Pass `True` for all boolean parameters to lift restrictions from a user.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat.
- **user_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target user. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **until_date** (*int, optional*) – Date when the user will be unbanned, unix time. If user is banned for more than 366 days or less than 30 seconds from the current time they are considered to be banned forever. Defaults to 0 (ban forever).
- **can_send_messages** (*bool, optional*) – Pass True, if the user can send text messages, contacts, locations and venues.
- **can_send_media_messages** (*bool, optional*) – Pass True, if the user can send audios, documents, photos, videos, video notes and voice notes, implies `can_send_messages`.
- **can_send_other_messages** (*bool, optional*) – Pass True, if the user can send animations, games, stickers and use inline bots, implies `can_send_media_messages`.
- **can_add_web_page_previews** (*bool, optional*) – Pass True, if the user may add web page previews to their messages, implies `can_send_media_messages`.
- **can_send_polls** (*bool, optional*) – Pass True, if the user can send polls, implies `can_send_media_messages`.
- **can_change_info** (*bool, optional*) – Pass True, if the user can change the chat title, photo and other settings.
- **can_invite_users** (*bool, optional*) – Pass True, if the user can invite new users to the chat.
- **can_pin_messages** (*bool, optional*) – Pass True, if the user can pin messages.

Returns *Chat* – On success, a chat object is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.promote_chat_member()`

Promote or demote a user in a supergroup or a channel.

You must be an administrator in the chat for this to work and must have the appropriate admin rights. Pass False for all boolean parameters to demote a user.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat.
- **user_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target user. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **can_change_info** (*bool, optional*) – Pass True, if the administrator can change chat title, photo and other settings.
- **can_post_messages** (*bool, optional*) – Pass True, if the administrator can create channel posts, channels only.
- **can_edit_messages** (*bool, optional*) – Pass True, if the administrator can edit messages of other users and can pin messages, channels only.
- **can_delete_messages** (*bool, optional*) – Pass True, if the administrator can delete messages of other users.
- **can_restrict_members** (*bool, optional*) – Pass True, if the administrator can restrict, ban or unban chat members.

- **can_invite_users** (*bool, optional*) – Pass True, if the administrator can invite new users to the chat.
- **can_pin_messages** (*bool, optional*) – Pass True, if the administrator can pin messages, supergroups only.
- **can_promote_members** (*bool, optional*) – Pass True, if the administrator can add new administrators with a subset of his own privileges or demote administrators that he has promoted, directly or indirectly (promoted by administrators that were appointed by him).

Returns *bool* – True on success.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.export_chat_invite_link()`

Generate a new invite link for a chat; any previously generated link is revoked.

You must be an administrator in the chat for this to work and have the appropriate admin rights.

Note: Each administrator in a chat generates their own invite links. Bots can't use invite links generated by other administrators. If you want your bot to work with invite links, it will need to generate its own link using this method – after this the link will become available to the bot via the `get_chat()` method. If your bot needs to generate a new invite link replacing its previous one, use this method again.

Parameters **chat_id** (*int | str*) – Unique identifier for the target chat or username of the target channel/supergroup (in the format `@username`).

Returns *str* – On success, the exported invite link is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.set_chat_photo()`

Set a new profile photo for the chat. Photos can't be changed for private chats. You must be an administrator in the chat for this to work and must have the appropriate admin rights.

Note: In regular groups (non-supergroups), this method will only work if the “All Members Are Admins” setting is off.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat.
- **photo** (*str*) – New chat photo. You can pass a *Photo* id or a file path to upload a new photo.

Returns *bool* – True on success.

Raises

- *RPCError* – In case of a Telegram RPC error.
- *ValueError* if a `chat_id` belongs to user.

`Client.delete_chat_photo()`

Delete a chat photo. Photos can't be changed for private chats. You must be an administrator in the chat for this to work and must have the appropriate admin rights.

Note: In regular groups (non-supergroups), this method will only work if the “All Members Are Admins” setting is off.

Parameters `chat_id` (int | str) – Unique identifier (int) or username (str) of the target chat.

Returns bool – True on success.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` if a `chat_id` belongs to user.

`Client.set_chat_title()`

Change the title of a chat. Titles can’t be changed for private chats. You must be an administrator in the chat for this to work and must have the appropriate admin rights.

Note: In regular groups (non-supergroups), this method will only work if the “All Members Are Admins” setting is off.

Parameters

- `chat_id` (int | str) – Unique identifier (int) or username (str) of the target chat.
- `title` (str) – New chat title, 1-255 characters.

Returns bool – True on success.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` – In case a chat id belongs to user.

`Client.set_chat_description()`

Change the description of a supergroup or a channel. You must be an administrator in the chat for this to work and must have the appropriate admin rights.

Parameters

- `chat_id` (int | str) – Unique identifier (int) or username (str) of the target chat.
- `description` (str) – New chat description, 0-255 characters.

Returns bool – True on success.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` if a `chat_id` doesn’t belong to a supergroup or a channel.

`Client.pin_chat_message()`

Pin a message in a group, channel or your own chat. You must be an administrator in the chat for this to work and must have the “can_pin_messages” admin right in the supergroup or “can_edit_messages” admin right in the channel.

Parameters

- `chat_id` (int | str) – Unique identifier (int) or username (str) of the target chat.

- **message_id** (int) – Identifier of a message to pin.
- **disable_notification** (bool) – Pass True, if it is not necessary to send a notification to all chat members about the new pinned message. Notifications are always disabled in channels.

Returns bool – True on success.

Raises RPCError – In case of a Telegram RPC error.

`Client.unpin_chat_message()`

Unpin a message in a group, channel or your own chat. You must be an administrator in the chat for this to work and must have the “can_pin_messages” admin right in the supergroup or “can_edit_messages” admin right in the channel.

Parameters **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat.

Returns bool – True on success.

Raises RPCError – In case of a Telegram RPC error.

`Client.get_chat()`

Get up to date information about a chat.

Information include current name of the user for one-on-one conversations, current username of a user, group or channel, etc.

Parameters **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat. Unique identifier for the target chat in form of a *t.me/joinchat/* link, identifier (int) or username of the target channel/supergroup (in the format @username).

Returns *Chat* | *ChatPreview* – On success, if you’ve already joined the chat, a chat object is returned, otherwise, a chat preview object is returned.

Raises

- RPCError – In case of a Telegram RPC error.
- ValueError – In case the chat invite link points to a chat you haven’t joined yet.

`Client.get_chat_member()`

Get information about one member of a chat.

Parameters

- **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat.
- **user_id** (int | str) – Unique identifier (int) or username (str) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (str).

Returns *ChatMember* – On success, a chat member is returned.

Raises RPCError – In case of a Telegram RPC error.

`Client.get_chat_members()`

Get a chunk of the members list of a chat.

You can get up to 200 chat members at once. A chat can be either a basic group, a supergroup or a channel. You must be admin to retrieve the members list of a channel (also known as “subscribers”). For a more convenient way of getting chat members see *iter_chat_members()*.

Parameters

- **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat.

- **offset** (*int, optional*) – Sequential number of the first member to be returned. Defaults to 0¹.
- **limit** (*int, optional*) – Limits the number of members to be retrieved. Defaults to 200, which is also the maximum server limit allowed per method call.
- **query** (*str, optional*) – Query string to filter members based on their display names and usernames. Defaults to "" (empty string)².
- **filter** (*str, optional*) – Filter used to select the kind of members you want to retrieve. Only applicable for supergroups and channels. It can be any of the followings: "all" - all kind of members, "kicked" - kicked (banned) members only, "restricted" - restricted members only, "bots" - bots only, "recent" - recent members only, "administrators" - chat administrators only. Defaults to "all".

Returns `ChatMembers` – On success, an object containing a list of chat members is returned.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` – In case you used an invalid filter or a chat id that belongs to a user.

`Client.get_chat_members_count()`

Get the number of members in a chat.

Parameters `chat_id` (*int | str*) – Unique identifier (int) or username (str) of the target chat.

Returns `int` – On success, the chat members count is returned.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` – In case a chat id belongs to user.

`Client.iter_chat_members()`

Iterate through the members of a chat sequentially.

This convenience method does the same as repeatedly calling `get_chat_members()` in a loop, thus saving you from the hassle of setting up boilerplate code. It is useful for getting the whole members list of a chat with a single call.

Parameters

- **chat_id** (*int | str*) – Unique identifier (int) or username (str) of the target chat.
- **limit** (*int, optional*) – Limits the number of members to be retrieved. By default, no limit is applied and all members are returned.
- **query** (*str, optional*) – Query string to filter members based on their display names and usernames. Defaults to "" (empty string).
- **filter** (*str, optional*) – Filter used to select the kind of members you want to retrieve. Only applicable for supergroups and channels. It can be any of the followings: "all" - all kind of members, "kicked" - kicked (banned) members only, "restricted" - restricted members only, "bots" - bots only, "recent" - recent members only, "administrators" - chat administrators only. Defaults to "all".

Returns `Generator` – A generator yielding `ChatMember` objects.

Raises `RPCError` – In case of a Telegram RPC error.

¹ Server limit: on supergroups, you can get up to 10,000 members for a single query and up to 200 members on channels.

² A query string is applicable only for "all", "kicked" and "restricted" filters only.

`Client.get_dialogs()`

Get a chunk of the user's dialogs.

You can get up to 100 dialogs at once. For a more convenient way of getting a user's dialogs see `iter_dialogs()`.

Parameters

- **offset_date** (int) – The offset date in Unix time taken from the top message of a *Dialog*. Defaults to 0. Valid for non-pinned dialogs only.
- **limit** (str, optional) – Limits the number of dialogs to be retrieved. Defaults to 100. Valid for non-pinned dialogs only.
- **pinned_only** (bool, optional) – Pass True if you want to get only pinned dialogs. Defaults to False.

Returns *Dialogs* – On success, an object containing a list of dialogs is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.iter_dialogs()`

Iterate through a user's dialogs sequentially.

This convenience method does the same as repeatedly calling `get_dialogs()` in a loop, thus saving you from the hassle of setting up boilerplate code. It is useful for getting the whole dialogs list with a single call.

Parameters

- **offset_date** (int) – The offset date in Unix time taken from the top message of a *Dialog*. Defaults to 0 (most recent dialog).
- **limit** (str, optional) – Limits the number of dialogs to be retrieved. By default, no limit is applied and all dialogs are returned.

Returns *Generator* – A generator yielding *Dialog* objects.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.get_dialogs_count()`

Get the total count of your dialogs.

pinned_only (bool, optional): Pass True if you want to count only pinned dialogs. Defaults to False.

Returns int – On success, the dialogs count is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.restrict_chat()`

Restrict a chat. Pass True for all boolean parameters to lift restrictions from a chat.

Parameters

- **chat_id** (int | str) – Unique identifier (int) or username (str) of the target chat.
- **can_send_messages** (bool, optional) – Pass True, if the user can send text messages, contacts, locations and venues.
- **can_send_media_messages** (bool, optional) – Pass True, if the user can send audios, documents, photos, videos, video notes and voice notes, implies `can_send_messages`.
- **can_send_other_messages** (bool, optional) – Pass True, if the user can send animations, games, stickers and use inline bots, implies `can_send_media_messages`.

- **can_add_web_page_previews** (*bool, optional*) – Pass True, if the user may add web page previews to their messages, implies `can_send_media_messages`.
- **can_send_polls** (*bool, optional*) – Pass True, if the user can send polls, implies `can_send_media_messages`.
- **can_change_info** (*bool, optional*) – Pass True, if the user can change the chat title, photo and other settings.
- **can_invite_users** (*bool, optional*) – Pass True, if the user can invite new users to the chat.
- **can_pin_messages** (*bool, optional*) – Pass True, if the user can pin messages.

Returns *Chat* – On success, a chat object is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.update_chat_username()`

Update a channel or a supergroup username.

To update your own username (for users only, not bots) you can use `update_username()`.

Parameters

- **chat_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat.
- **username** (*str | None*) – Username to set. Pass "" (empty string) or None to remove the username.

Returns *bool* – True on success.

Raises

- *RPCError* – In case of a Telegram RPC error.
- *ValueError* – In case a chat id belongs to a user or chat.

`Client.get_me()`

Get your own user identity.

Returns *User* – Basic information about the user or bot.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.get_users()`

Get information about a user. You can retrieve up to 200 users at once.

Parameters **user_ids** (*iterable*) – A list of User identifiers (*id* or *username*) or a single user *id/username*. For a contact that exists in your Telegram address book you can use his phone number (*str*). Iterators and Generators are also accepted.

Returns *User* | List of *User* – In case *user_ids* was an integer or string the single requested user is returned, otherwise, in case *user_ids* was an iterable a list of users is returned, even if the iterable contained one item only.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.get_user_profile_photos()`

Get a list of profile pictures for a user.

Parameters

- **user_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use "me" or "self". For a contact that exists in your Telegram address book you can use his phone number (*str*).

- **offset** (*int, optional*) – Sequential number of the first photo to be returned. By default, all photos are returned.
- **limit** (*int, optional*) – Limits the number of photos to be retrieved. Values between 1—100 are accepted. Defaults to 100.

Returns *UserProfilePhotos* – On success, an object containing a list of the profile photos is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.get_user_profile_photos_count()`

Get the total count of profile pictures for a user.

Parameters **user_id** (*int | str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).

Returns *int* – On success, the user profile photos count is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.set_user_profile_photo()`

Set a new profile photo.

This method only works for Users. Bots profile photos must be set using BotFather.

Parameters **photo** (*str*) – Profile photo to set. Pass a file path as string to upload a new photo that exists on your local machine.

Returns *bool* – True on success.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.delete_user_profile_photos()`

Delete your own profile photos.

Parameters **id** (*str | list*) – A single *Photo* id as string or multiple ids as list of strings for deleting more than one photos at once.

Returns *bool* – True on success.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.update_username()`

Update your own username.

This method only works for users, not bots. Bot usernames must be changed via Bot Support or by recreating them from scratch using BotFather. To update a channel or supergroup username you can use `update_chat_username()`.

Parameters **username** (*str | None*) – Username to set. “” (empty string) or *None* to remove it.

Returns *bool* – True on success.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.add_contacts()`

Add contacts to your Telegram address book.

Parameters **contacts** (List of *InputPhoneContact*) – The contact list to be added

Raises *RPCError* – In case of a Telegram RPC error.

`Client.get_contacts()`

Get contacts from your Telegram address book.

Returns List of *User* – On success, a list of users is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.get_contacts_count()`

Get the total count of contacts from your Telegram address book.

Returns *int* – On success, the contacts count is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.delete_contacts()`

Delete contacts from your Telegram address book.

Parameters *ids* (List of *int*) – A list of unique identifiers for the target users. Can be an ID (*int*), a username (*string*) or phone number (*string*).

Returns *bool* – True on success.

Raises *RPCError* – In case of a Telegram RPC error.

`Client.enable_cloud_password()`

Enable the Two-Step Verification security feature (Cloud Password) on your account.

This password will be asked when you log-in on a new device in addition to the SMS code.

Parameters

- **password** (*str*) – Your password.
- **hint** (*str*, *optional*) – A password hint.
- **email** (*str*, *optional*) – Recovery e-mail.

Returns *bool* – True on success.

Raises

- *RPCError* – In case of a Telegram RPC error.
- *ValueError* – In case there is already a cloud password enabled.

`Client.change_cloud_password()`

Change your Two-Step Verification password (Cloud Password) with a new one.

Parameters

- **current_password** (*str*) – Your current password.
- **new_password** (*str*) – Your new password.
- **new_hint** (*str*, *optional*) – A new password hint.

Returns *bool* – True on success.

Raises

- *RPCError* – In case of a Telegram RPC error.
- *ValueError* – In case there is no cloud password to change.

`Client.remove_cloud_password()`

Turn off the Two-Step Verification security feature (Cloud Password) on your account.

Parameters **password** (*str*) – Your current password.

Returns *bool* – True on success.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` – In case there is no cloud password to remove.

`Client.get_inline_bot_results()`

Get bot results via inline queries. You can then send a result using `send_inline_bot_result`

Parameters

- **bot** (`int` | `str`) – Unique identifier of the inline bot you want to get results from. You can specify a `@username` (`str`) or a bot ID (`int`).
- **query** (`str`) – Text of the query (up to 512 characters).
- **offset** (`str`, *optional*) – Offset of the results to be returned.
- **latitude** (`float`, *optional*) – Latitude of the location. Useful for location-based results only.
- **longitude** (`float`, *optional*) – Longitude of the location. Useful for location-based results only.

Returns `BotResults` – On Success.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `TimeoutError` – In case the bot fails to answer within 10 seconds.

`Client.send_inline_bot_result()`

Send an inline bot result. Bot results can be retrieved using `get_inline_bot_results`

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **query_id** (`int`) – Unique identifier for the answered query.
- **result_id** (`str`) – Unique identifier for the result that was chosen.
- **disable_notification** (`bool`, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (`bool`, *optional*) – If the message is a reply, ID of the original message.
- **hide_via** (`bool`) – Sends the message with *via @bot* hidden.

Returns `Message` – On success, the sent inline result message is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.answer_callback_query()`

Send answers to callback queries sent from inline keyboards. The answer will be displayed to the user as a notification at the top of the chat screen or as an alert.

Parameters

- **callback_query_id** (`str`) – Unique identifier for the query to be answered.
- **text** (`str`) – Text of the notification. If not specified, nothing will be shown to the user, 0-200 characters.

- **show_alert** (*bool*) – If true, an alert will be shown by the client instead of a notification at the top of the chat screen. Defaults to False.
- **url** (*str*) – URL that will be opened by the user’s client. If you have created a Game and accepted the conditions via @Botfather, specify the URL that opens your game – note that this will only work if the query comes from a `callback_game` button. Otherwise, you may use links like `t.me/your_bot?start=XXXX` that open your bot with a parameter.
- **cache_time** (*int*) – The maximum amount of time in seconds that the result of the callback query may be cached client-side. Telegram apps will support caching starting in version 3.14. Defaults to 0.

Returns *bool* – True, on success.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.answer_inline_query()`

Send answers to an inline query. No more than 50 results per query are allowed.

Parameters

- **inline_query_id** (*str*) – Unique identifier for the answered query.
- **results** (List of *InlineQueryResult*) – A list of results for the inline query.
- **cache_time** (*int, optional*) – The maximum amount of time in seconds that the result of the inline query may be cached on the server. Defaults to 300.
- **is_personal** (*bool, optional*) – Pass True, if results may be cached on the server side only for the user that sent the query. By default, results may be returned to any user who sends the same query.
- **next_offset** (*str, optional*) – Pass the offset that a client should send in the next query with the same text to receive more results. Pass an empty string if there are no more results or if you don’t support pagination. Offset length can’t exceed 64 bytes.
- **switch_pm_text** (*str, optional*) – If passed, clients will display a button with specified text that switches the user to a private chat with the bot and sends the bot a start message with the parameter `switch_pm_parameter`
- **switch_pm_parameter** (*str, optional*) – Deep-linking parameter for the /start message sent to the bot when user presses the switch button. 1-64 characters, only A-Z, a-z, 0-9, _ and - are allowed.

Example: An inline bot that sends YouTube videos can ask the user to connect the bot to their YouTube account to adapt search results accordingly. To do this, it displays a “Connect your YouTube account” button above the results, or even before showing any. The user presses the button, switches to a private chat with the bot and, in doing so, passes a start parameter that instructs the bot to return an oauth link. Once done, the bot can offer a `switch_inline` button so that the user can easily return to the chat where they wanted to use the bot’s inline capabilities.

Returns *bool* – True, on success.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.request_callback_answer()`

Request a callback answer from bots. This is the equivalent of clicking an inline button containing callback data.

Parameters

- **chat_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **message_id** (*int*) – The message id the inline keyboard is attached on.
- **callback_data** (*str* | *bytes*) – Callback data associated with the inline button you want to get the answer from.
- **timeout** (*int*, *optional*) – Timeout in seconds.

Returns The answer containing info useful for clients to display a notification at the top of the chat screen or as an alert.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `TimeoutError` – In case the bot fails to answer within 10 seconds.

`Client.send_game()`

Send a game.

Parameters

- **chat_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **game_short_name** (*str*) – Short name of the game, serves as the unique identifier for the game. Set up your games via Botfather.
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int*, *optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup*, *optional*) – An object for an inline keyboard. If empty, one ‘Play game_title’ button will be shown automatically. If not empty, the first button must launch the game.

Returns *Message* – On success, the sent game message is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.set_game_score()`

Set the score of the specified user in a game.

Parameters

- **user_id** (*int* | *str*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (*str*).
- **score** (*int*) – New score, must be non-negative.
- **force** (*bool*, *optional*) – Pass True, if the high score is allowed to decrease. This can be useful when fixing mistakes or banning cheaters.
- **disable_edit_message** (*bool*, *optional*) – Pass True, if the game message should not be automatically edited to include the current scoreboard.
- **chat_id** (*int* | *str*, *optional*) – Unique identifier (*int*) or username (*str*) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For

a contact that exists in your Telegram address book you can use his phone number (str). Required if `inline_message_id` is not specified.

- **message_id** (int, *optional*) – Identifier of the sent message. Required if `inline_message_id` is not specified.

Returns `Message` | `bool` – On success, if the message was sent by the bot, the edited message is returned, `True` otherwise.

Raises `RPCError` – In case of a Telegram RPC error.

`Client.get_game_high_scores()`

Get data for high score tables.

Parameters

- **user_id** (int | str) – Unique identifier (int) or username (str) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (str).
- **chat_id** (int | str, *optional*) – Unique identifier (int) or username (str) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (str). Required if `inline_message_id` is not specified.
- **message_id** (int, *optional*) – Identifier of the sent message. Required if `inline_message_id` is not specified.

Returns `GameHighScores` – On success.

Raises `RPCError` – In case of a Telegram RPC error.

4.10 Available Types

All Pyrogram types listed here are accessible through the main package directly.

```
from pyrogram import User, Message, ...
```

Note: **Optional** fields may not exist when irrelevant – i.e.: they will contain the value of `None` and aren’t shown when, for example, using `print()`.

4.10.1 Users & Chats

<code>User</code>	A Telegram user or bot.
<code>UserStatus</code>	A User status (Last Seen privacy).
<code>Chat</code>	A chat.
<code>ChatPreview</code>	A chat preview.
<code>ChatPhoto</code>	A chat photo.
<code>ChatMember</code>	Contains information about one member of a chat.
<code>ChatMembers</code>	Contains information about the members list of a chat.
<code>ChatPermissions</code>	A chat default permissions and a single member permissions within a chat.
<code>Dialog</code>	A user’s dialog.

Continued on next page

Table 8 – continued from previous page

<i>Dialogs</i>	Contains a user's dialogs chunk.
----------------	----------------------------------

4.10.2 Messages & Media

<i>Message</i>	A message.
<i>Messages</i>	Contains a chat's messages.
<i>MessageEntity</i>	One special entity in a text message.
<i>Photo</i>	A Photo.
<i>PhotoSize</i>	One size of a photo or a file/sticker thumbnail.
<i>UserProfilePhotos</i>	Contains a user's profile pictures.
<i>Audio</i>	An audio file to be treated as music by the Telegram clients.
<i>Document</i>	A generic file (as opposed to photos, voice messages, audio files, ...).
<i>Animation</i>	An animation file (GIF or H.264/MPEG-4 AVC video without sound).
<i>Video</i>	A video file.
<i>Voice</i>	A voice note.
<i>VideoNote</i>	A video note.
<i>Contact</i>	A phone contact.
<i>Location</i>	A point on the map.
<i>Venue</i>	A venue.
<i>Sticker</i>	A sticker.
<i>Game</i>	A game.
<i>Poll</i>	A Poll.
<i>PollOption</i>	Contains information about one answer option in a poll.

4.10.3 Keyboards

<i>ReplyKeyboardMarkup</i>	A custom keyboard with reply options.
<i>KeyboardButton</i>	One button of the reply keyboard.
<i>ReplyKeyboardRemove</i>	Object used to tell clients to remove a bot keyboard.
<i>InlineKeyboardMarkup</i>	An inline keyboard that appears right next to the message it belongs to.
<i>InlineKeyboardButton</i>	One button of an inline keyboard.
<i>ForceReply</i>	Object used to force clients to show a reply interface.
<i>CallbackQuery</i>	An incoming callback query from a callback button in an inline keyboard.
<i>GameHighScore</i>	One row of the high scores table for a game.
<i>GameHighScores</i>	The high scores table for a game.
<i>CallbackGame</i>	Placeholder, currently holds no information.

4.10.4 Input Media

<i>InputMedia</i>	Content of a media message to be sent.
<i>InputMediaPhoto</i>	A photo to be sent inside an album.
<i>InputMediaVideo</i>	A video to be sent inside an album.

Continued on next page

Table 11 – continued from previous page

<i>InputMediaAudio</i>	An audio to be sent inside an album.
<i>InputMediaAnimation</i>	An animation file (GIF or H.264/MPEG-4 AVC video without sound) to be sent inside an album.
<i>InputMediaDocument</i>	A generic file to be sent inside an album.
<i>InputPhoneContact</i>	A Phone Contact to be added in your Telegram address book.

4.10.5 Inline Mode

<i>InlineQuery</i>	An incoming inline query.
<i>InlineQueryResult</i>	One result of an inline query.
<i>InlineQueryResultArticle</i>	Link to an article or web page.

4.10.6 InputMessageContent

<i>InputMessageContent</i>	Content of a message to be sent as a result of an inline query.
<i>InputTextMessageContent</i>	Content of a text message to be sent as the result of an inline query.

class pyrogram.**User**
A Telegram user or bot.

Parameters

- **id** (*int*) – Unique identifier for this user or bot.
- **is_self** (*bool*) – True, if this user is you yourself.
- **is_contact** (*bool*) – True, if this user is in your contacts.
- **is_mutual_contact** (*bool*) – True, if you both have each other's contact.
- **is_deleted** (*bool*) – True, if this user is deleted.
- **is_bot** (*bool*) – True, if this user is a bot.
- **first_name** (*str*) – User's or bot's first name.
- **status** (*UserStatus*, *optional*) – User's Last Seen status. Empty for bots.
- **last_name** (*str*, *optional*) – User's or bot's last name.
- **username** (*str*, *optional*) – User's or bot's username.
- **language_code** (*str*, *optional*) – IETF language tag of the user's language.
- **phone_number** (*str*, *optional*) – User's phone number.
- **photo** (*ChatPhoto*, *optional*) – User's or bot's current profile photo. Suitable for downloads only.
- **restriction_reason** (*str*, *optional*) – The reason why this bot might be unavailable to some users.

class pyrogram.**UserStatus**
A User status (Last Seen privacy).

Note: You won't see exact last seen timestamps for people with whom you don't share your own. Instead, you get "recently", "within_week", "within_month" or "long_time_ago" fields set.

Parameters

- **user_id** (*int*) – User's id.
- **online** (*bool, optional*) – True if the user is online in this very moment, None otherwise. If True, the "date" field will be also set containing the online expiration date (i.e.: the date when a user will automatically go offline in case of no action by his client).
- **offline** (*bool, optional*) – True if the user is offline in this moment and has the Last Seen privacy setting public, None otherwise. If True, the "date" field will be also set containing the last seen date (i.e.: the date when a user was online the last time).
- **date** (*int, optional*) – Exact date in unix time. Available only in case "online" or "offline" equals to True.
- **recently** (*bool, optional*) – True for users with hidden Last Seen privacy that have been online between 1 second and 2-3 days ago, None otherwise.
- **within_week** (*bool, optional*) – True for users with hidden Last Seen privacy that have been online between 2-3 and seven days ago, None otherwise.
- **within_month** (*bool, optional*) – True for users with hidden Last Seen privacy that have been online between 6-7 days and a month ago, None otherwise.
- **long_time_ago** (*bool, optional*) – True for users with hidden Last Seen privacy that have been online more than a month ago (this is also always shown to blocked users), None otherwise.

class pyrogram.Chat

A chat.

Parameters

- **id** (*int*) – Unique identifier for this chat.
- **type** (*str*) – Type of chat, can be either "private", "bot", "group", "supergroup" or "channel".
- **title** (*str, optional*) – Title, for supergroups, channels and basic group chats.
- **username** (*str, optional*) – Username, for private chats, bots, supergroups and channels if available.
- **first_name** (*str, optional*) – First name of the other party in a private chat, for private chats and bots.
- **last_name** (*str, optional*) – Last name of the other party in a private chat, for private chats.
- **photo** (*ChatPhoto, optional*) – Chat photo. Suitable for downloads only.
- **description** (*str, optional*) – Bio, for private chats and bots or description for groups, supergroups and channels. Returned only in `get_chat()`.
- **invite_link** (*str, optional*) – Chat invite link, for groups, supergroups and channels. Returned only in `get_chat()`.

- **pinned_message** (*Message, optional*) – Pinned message, for groups, supergroups channels and own chat. Returned only in `get_chat()`.
- **sticker_set_name** (*str, optional*) – For supergroups, name of group sticker set. Returned only in `get_chat()`.
- **can_set_sticker_set** (*bool, optional*) – True, if the group sticker set can be changed by you. Returned only in `get_chat()`.
- **members_count** (*int, optional*) – Chat members count, for groups, supergroups and channels only.
- **restriction_reason** (*str, optional*) – The reason why this chat might be unavailable to some users.
- **permissions** (*ChatPermissions optional*) – Information about the chat default permissions, for groups and supergroups.

class pyrogram.ChatPreview

A chat preview.

Parameters

- **title** (*str*) – Title of the chat.
- **photo** (*ChatPhoto, optional*) – Chat photo. Suitable for downloads only.
- **type** (*str*) – Type of chat, can be either, “group”, “supergroup” or “channel”.
- **members_count** (*int*) – Chat members count.
- **members** (List of *User, optional*) – Preview of some of the chat members.

class pyrogram.ChatPhoto

A chat photo.

Parameters

- **small_file_id** (*str*) – Unique file identifier of small (160x160) chat photo. This `file_id` can be used only for photo download.
- **big_file_id** (*str*) – Unique file identifier of big (640x640) chat photo. This `file_id` can be used only for photo download.

class pyrogram.ChatMember

Contains information about one member of a chat.

Parameters

- **user** (*User*) – Information about the user.
- **status** (*str*) – The member’s status in the chat. Can be “creator”, “administrator”, “member”, “restricted”, “left” or “kicked”.
- **date** (*int, optional*) – Date when the user joined, unix time. Not available for creator.
- **is_member** (*bool, optional*) – Restricted only. True, if the user is a member of the chat at the moment of the request.
- **invited_by** (*User, optional*) – Administrators and self member only. Information about the user who invited this member. In case the user joined by himself this will be the same as “user”.
- **promoted_by** (*User, optional*) – Administrators only. Information about the user who promoted this member as administrator.

- **restricted_by** (*User, optional*) – Restricted and kicked only. Information about the user who restricted or kicked this member.
- **permissions** (*ChatPermissions optional*) – Administrators, restricted and kicked members only. Information about the member permissions.

class pyrogram.ChatMembers

Contains information about the members list of a chat.

Parameters

- **total_count** (*int*) – Total number of members the chat has.
- **chat_members** (List of *ChatMember*) – Requested chat members.

class pyrogram.ChatPermissions

A chat default permissions and a single member permissions within a chat.

Some permissions make sense depending on the context: default chat permissions, restricted/kicked member or administrators in groups or channels.

Parameters

- **until_date** (*int, optional*) – Applicable to restricted and kicked members only. Date when user restrictions will be lifted, unix time. 0 means the restrictions will never be lifted (user restricted forever).
- **can_be_edited** (*bool, optional*) – Applicable to administrators only. True, if you are allowed to edit administrator privileges of the user.
- **can_change_info** (*bool, optional*) – Applicable to default chat permissions in private groups and administrators in public groups only. True, if the chat title, photo and other settings can be changed.
- **can_post_messages** (*bool, optional*) – Applicable to channel administrators only. True, if the administrator can post messages in the channel, channels only.
- **can_edit_messages** (*bool, optional*) – Applicable to channel administrators only. True, if the administrator can edit messages of other users and can pin messages, channels only.
- **can_delete_messages** (*bool, optional*) – Applicable to administrators only. True, if the administrator can delete messages of other users.
- **can_restrict_members** (*bool, optional*) – Applicable to administrators only. True, if the administrator can restrict, ban or unban chat members.
- **can_invite_users** (*bool, optional*) – Applicable to default chat permissions and administrators only. True, if new users can be invited to the chat.
- **can_pin_messages** (*bool, optional*) – Applicable to default chat permissions in private groups and administrators in public groups only. True, if messages can be pinned, supergroups only.
- **can_promote_members** (*bool, optional*) – Applicable to administrators only. True, if the administrator can add new administrators with a subset of his own privileges or demote administrators that he has promoted, directly or indirectly (promoted by administrators that were appointed by the user).
- **can_send_messages** (*bool, optional*) – Applicable to default chat permissions and restricted members only. True, if text messages, contacts, locations and venues can be sent.

- **can_send_media_messages** (*bool, optional*) – Applicable to default chat permissions and restricted members only. True, if audios, documents, photos, videos, video notes and voice notes can be sent, implies can_send_messages.
- **can_send_other_messages** (*bool, optional*) – Applicable to default chat permissions and restricted members only. True, if animations, games, stickers and inline bot results can be sent, implies can_send_media_messages.
- **can_add_web_page_previews** (*bool, optional*) – Applicable to default chat permissions and restricted members only. True, if web page previews can be attached to text messages, implies can_send_media_messages.
- **can_send_polls** (*bool, optional*) – Applicable to default chat permissions and restricted members only. True, if polls can be sent, implies can_send_media_messages.

class pyrogram.Dialog

A user's dialog.

Parameters

- **chat** (*Chat*) – Conversation the dialog belongs to.
- **top_message** (*Message*) – The last message sent in the dialog at this time.
- **unread_messages_count** (*int*) – Amount of unread messages in this dialog.
- **unread_mentions_count** (*int*) – Amount of unread messages containing a mention in this dialog.
- **unread_mark** (*bool*) – True, if the dialog has the unread mark set.
- **is_pinned** (*bool*) – True, if the dialog is pinned.

class pyrogram.Dialogs

Contains a user's dialogs chunk.

Parameters

- **total_count** (*int*) – Total number of dialogs the user has.
- **dialogs** (List of *Dialog*) – Requested dialogs.

class pyrogram.Message

A message.

Parameters

- **message_id** (*int*) – Unique message identifier inside this chat.
- **date** (*int, optional*) – Date the message was sent in Unix time.
- **chat** (*Chat, optional*) – Conversation the message belongs to.
- **from_user** (*User, optional*) – Sender, empty for messages sent to channels.
- **forward_from** (*User, optional*) – For forwarded messages, sender of the original message.
- **forward_sender_name** (*str, optional*) – For messages forwarded from users who have hidden their accounts, name of the user.
- **forward_from_chat** (*Chat, optional*) – For messages forwarded from channels, information about the original channel.
- **forward_from_message_id** (*int, optional*) – For messages forwarded from channels, identifier of the original message in the channel.

- **forward_signature** (*str, optional*) – For messages forwarded from channels, signature of the post author if present.
- **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.
- **mentioned** (*bool, optional*) – The message contains a mention.
- **empty** (*bool, optional*) – The message is empty. A message can be empty in case it was deleted or you tried to retrieve a message that doesn't exist yet.
- **service** (*bool, optional*) – The message is a service message. A service message has one and only one of these fields set: left_chat_member, new_chat_title, new_chat_photo, delete_chat_photo, group_chat_created, supergroup_chat_created, channel_chat_created, migrate_to_chat_id, migrate_from_chat_id, pinned_message.
- **media** (*bool optional*) – The message is a media message. A media message has one and only one of these fields set: audio, document, photo, sticker, video, animation, voice, video_note, contact, location, venue.
- **edit_date** (*int, optional*) – Date the message was last edited in Unix time.
- **media_group_id** (*str, optional*) – The unique identifier of a media message group this message belongs to.
- **author_signature** (*str, optional*) – Signature of the post author for messages in channels.
- **text** (*str, optional*) – For text messages, the actual UTF-8 text of the message, 0-4096 characters. If the message contains entities (bold, italic, ...) you can access *text.markdown* or *text.html* to get the marked up message text. In case there is no entity, the fields will contain the same text as *text*.
- **entities** (List of *MessageEntity, optional*) – For text messages, special entities like usernames, URLs, bot commands, etc. that appear in the text.
- **caption_entities** (List of *MessageEntity, optional*) – For messages with a caption, special entities like usernames, URLs, bot commands, etc. that appear in the caption.
- **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** (*Photo, optional*) – Message is a photo, information about the photo.
- **sticker** (*Sticker, optional*) – Message is a sticker, information about the sticker.
- **animation** (*Animation, optional*) – Message is an animation, information about the animation.
- **game** (*Game, optional*) – Message is a game, information about the game.
- **video** (*Video, optional*) – Message is a video, information about the video.
- **voice** (*Voice, optional*) – Message is a voice message, information about the file.
- **video_note** (*VideoNote, optional*) – Message is a video note, information about the video message.
- **caption** (*str, optional*) – Caption for the audio, document, photo, video or voice, 0-1024 characters. If the message contains caption entities (bold, italic, ...) you can access

caption.markdown or *caption.html* to get the marked up caption text. In case there is no caption entity, the fields will contain the same text as *caption*.

- **contact** (*Contact*, *optional*) – Message is a shared contact, information about the contact.
- **location** (*Location*, *optional*) – Message is a shared location, information about the location.
- **venue** (*Venue*, *optional*) – Message is a venue, information about the venue.
- **web_page** (*bool*, *optional*) – Message was sent with a webpage preview. **Note:** Support for web pages is still basic; a simple boolean is set in case the message contains a web page preview. In future versions this property could turn into a full web page object that contains more details.
- **poll** (*Poll*, *optional*) – Message is a native poll, information about the poll.
- **new_chat_members** (List of *User*, *optional*) – New members that were added to the group or supergroup and information about them (the bot itself may be one of these members).
- **left_chat_member** (*User*, *optional*) – A member was removed from the group, information about them (this member may be the bot itself).
- **new_chat_title** (*str*, *optional*) – A chat title was changed to this value.
- **new_chat_photo** (*Photo*, *optional*) – A chat photo was change to this value.
- **delete_chat_photo** (*bool*, *optional*) – Service message: the chat photo was deleted.
- **group_chat_created** (*bool*, *optional*) – Service message: the group has been created.
- **supergroup_chat_created** (*bool*, *optional*) – Service message: the supergroup has been created. This field can't be received in a message coming through updates, because bot can't be a member of a supergroup when it is created. It can only be found in *reply_to_message* if someone replies to a very first message in a directly created supergroup.
- **channel_chat_created** (*bool*, *optional*) – Service message: the channel has been created. This field can't be received in a message coming through updates, because bot can't be a member of a channel when it is created. It can only be found in *reply_to_message* if someone replies to a very first message in a channel.
- **migrate_to_chat_id** (*int*, *optional*) – The group has been migrated to a supergroup with the specified identifier. This number may be greater than 32 bits and some programming languages may have difficulty/silent defects in interpreting it. But it is smaller than 52 bits, so a signed 64 bit integer or double-precision float type are safe for storing this identifier.
- **migrate_from_chat_id** (*int*, *optional*) – The supergroup has been migrated from a group with the specified identifier. This number may be greater than 32 bits and some programming languages may have difficulty/silent defects in interpreting it. But it is smaller than 52 bits, so a signed 64 bit integer or double-precision float type are safe for storing this identifier.
- **pinned_message** (*Message*, *optional*) – Specified message was pinned. Note that the Message object in this field will not contain further *reply_to_message* fields even if it is itself a reply.
- **game_high_score** (*GameHighScore*, *optional*) – The game score for a user. The *reply_to_message* field will contain the game Message.

- **views** (*int, optional*) – Channel post views.
- **via_bot** (*User*) – The information of the bot that generated the message from an inline query of a user.
- **outgoing** (*bool, optional*) – Whether the message is incoming or outgoing. Messages received from other chats are incoming (*outgoing* is False). Messages sent from yourself to other chats are outgoing (*outgoing* is True). An exception is made for your own personal chat; messages sent there will be incoming.
- **matches** (List of regex Matches, *optional*) – A list containing all [Match Objects](#) that match the text of this message. Only applicable when using *Filters.regex*.
- **command** (List of *str, optional*) – A list containing the command and its arguments, if any. E.g.: “/start 1 2 3” would produce [“start”, “1”, “2”, “3”]. Only applicable when using *Filters.command*.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

class pyrogram.**Messages**

Contains a chat’s messages.

Parameters

- **total_count** (*int*) – Total number of messages the target chat has.
- **messages** (List of *Message*) – Requested messages.

class pyrogram.**MessageEntity**

One special entity in a text message. For example, hashtags, usernames, URLs, etc.

Parameters

- **type** (*str*) – Type of the entity. Can be “mention” (@username), “hashtag”, “cashtag”, “bot_command”, “url”, “email”, “phone_number”, “bold” (bold text), italic (italic text), “code” (monowidth string), “pre” (monowidth block), “text_link” (for clickable text URLs), “text_mention” (for users without usernames).
- **offset** (*int*) – Offset in UTF-16 code units to the start of the entity.
- **length** (*int*) – Length of the entity in UTF-16 code units.
- **url** (*str, optional*) – For “text_link” only, url that will be opened after user taps on the text.
- **user** (*User, optional*) – For “text_mention” only, the mentioned user.

class pyrogram.**Photo**

A Photo.

Parameters

- **id** (*str*) – Unique identifier for this photo.
- **date** (*int*) – Date the photo was sent in Unix time.
- **sizes** (List of *PhotoSize*) – Available sizes of this photo.

class pyrogram.**PhotoSize**

One size of a photo or a file/sticker thumbnail.

Parameters

- **file_id**(str) – Unique identifier for this file.
- **width**(int) – Photo width.
- **height**(int) – Photo height.
- **file_size**(int) – File size.

class pyrogram.**UserProfilePhotos**

Contains a user's profile pictures.

Parameters

- **total_count**(int) – Total number of profile pictures the target user has.
- **photos**(List of *Photo*) – Requested profile pictures.

class pyrogram.**Audio**

An audio file to be treated as music by the Telegram clients.

Parameters

- **file_id**(str) – Unique identifier for this file.
- **duration**(int) – Duration of the audio in seconds as defined by sender.
- **thumb**(*PhotoSize*, optional) – Thumbnail of the music file album cover.
- **file_name**(str, optional) – Audio file name.
- **mime_type**(str, optional) – MIME type of the file as defined by sender.
- **file_size**(int, optional) – File size.
- **date**(int, optional) – Date the audio was sent in Unix time.
- **performer**(str, optional) – Performer of the audio as defined by sender or by audio tags.
- **title**(str, optional) – Title of the audio as defined by sender or by audio tags.

class pyrogram.**Document**

A generic file (as opposed to photos, voice messages, audio files, ...).

Parameters

- **file_id**(str) – Unique file identifier.
- **thumb**(*PhotoSize*, optional) – 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.
- **date**(int, optional) – Date the document was sent in Unix time.

class pyrogram.**Animation**

An animation file (GIF or H.264/MPEG-4 AVC video without sound).

Parameters

- **file_id**(str) – Unique identifier for this file.
- **width**(int) – Animation width as defined by sender.
- **height**(int) – Animation height as defined by sender.
- **duration**(int) – Duration of the animation in seconds as defined by sender.

- **thumb** (*PhotoSize, optional*) – Animation thumbnail.
- **file_name** (*str, optional*) – Animation file name.
- **mime_type** (*str, optional*) – Mime type of a file as defined by sender.
- **file_size** (*int, optional*) – File size.
- **date** (*int, optional*) – Date the animation was sent in Unix time.

class pyrogram.Video
A video file.

Parameters

- **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, optional*) – Video thumbnail.
- **file_name** (*str, optional*) – Video file name.
- **mime_type** (*str, optional*) – Mime type of a file as defined by sender.
- **supports_streaming** (*bool, optional*) – True, if the video was uploaded with streaming support.
- **file_size** (*int, optional*) – File size.
- **date** (*int, optional*) – Date the video was sent in Unix time.

class pyrogram.Voice
A voice note.

Parameters

- **file_id** (*str*) – Unique identifier for this file.
- **duration** (*int*) – Duration of the audio in seconds as defined by sender.
- **waveform** (*bytes, optional*) – Voice waveform.
- **mime_type** (*str, optional*) – MIME type of the file as defined by sender.
- **file_size** (*int, optional*) – File size.
- **date** (*int, optional*) – Date the voice was sent in Unix time.

class pyrogram.VideoNote
A video note.

Parameters

- **file_id** (*str*) – Unique identifier for this file.
- **length** (*int*) – Video width and height as defined by sender.
- **duration** (*int*) – Duration of the video in seconds as defined by sender.
- **thumb** (*PhotoSize, optional*) – Video thumbnail.
- **mime_type** (*str, optional*) – MIME type of the file as defined by sender.
- **file_size** (*int, optional*) – File size.

- **date** (*int, optional*) – Date the video note was sent in Unix time.

class pyrogram.**Contact**

A phone contact.

Parameters

- **phone_number** (*str*) – Contact’s phone number.
- **first_name** (*str*) – Contact’s first name.
- **last_name** (*str, optional*) – Contact’s last name.
- **user_id** (*int, optional*) – Contact’s user identifier in Telegram.
- **vcard** (*str, optional*) – Additional data about the contact in the form of a vCard.

class pyrogram.**Location**

A point on the map.

Parameters

- **longitude** (*float*) – Longitude as defined by sender.
- **latitude** (*float*) – Latitude as defined by sender.

class pyrogram.**Venue**

A venue.

Parameters

- **location** (*Location*) – Venue location.
- **title** (*str*) – Name of the venue.
- **address** (*str*) – Address of the venue.
- **foursquare_id** (*str, optional*) – Foursquare identifier of the venue.
- **foursquare_type** (*str, optional*) – Foursquare type of the venue. (For example, “arts_entertainment/default”, “arts_entertainment/aquarium” or “food/icecream”).

class pyrogram.**Sticker**

A sticker.

Parameters

- **file_id** (*str*) – Unique identifier for this file.
- **width** (*int*) – Sticker width.
- **height** (*int*) – Sticker height.
- **thumb** (*PhotoSize, optional*) – Sticker thumbnail in the .webp or .jpg format.
- **file_name** (*str, optional*) – Sticker file name.
- **mime_type** (*str, optional*) – MIME type of the file as defined by sender.
- **file_size** (*int, optional*) – File size.
- **date** (*int, optional*) – Date the sticker was sent in Unix time.
- **emoji** (*str, optional*) – Emoji associated with the sticker.
- **set_name** (*str, optional*) – Name of the sticker set to which the sticker belongs.

class pyrogram.**Game**

A game. Use BotFather to create and edit games, their short names will act as unique identifiers.

Parameters

- **id** (*int*) – Unique identifier of the game.
- **title** (*str*) – Title of the game.
- **short_name** (*str*) – Unique short name of the game.
- **description** (*str*) – Description of the game.
- **photo** (*Photo*) – Photo that will be displayed in the game message in chats.
- **animation** (*Animation, optional*) – Animation that will be displayed in the game message in chats. Upload via BotFather.

```
class pyrogram.Poll
    A Poll.
```

Parameters

- **id** (*str*) – Unique poll identifier.
- **question** (*str*) – Poll question, 1-255 characters.
- **options** (List of *PollOption*) – List of poll options.
- **is_closed** (*bool*) – True, if the poll is closed.
- **total_voters** (*int*) – Total count of voters for this poll.
- **chosen_option** (*int, optional*) – Index of your chosen option (0-9), None in case you haven't voted yet.

```
class pyrogram.PollOption
    Contains information about one answer option in a poll.
```

Parameters

- **text** (*str*) – Option text, 1-100 characters.
- **voter_count** (*int*) – Number of users that voted for this option. Equals to 0 until you vote.
- **data** (*bytes*) – The data this poll option is holding.

```
class pyrogram.ReplyKeyboardMarkup
    A custom keyboard with reply options.
```

Parameters

- **keyboard** (List of List of *KeyboardButton*) – List of button rows, each represented by a List of KeyboardButton objects.
- **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. The keyboard will still be available, but clients will automatically display the usual letter-keyboard in the chat – the user can press a special button in the input field to see the custom keyboard again. 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.

class pyrogram.KeyboardButton

One button of the reply keyboard. For simple text buttons String can be used instead of this object to specify text of the button. Optional fields are mutually exclusive.

Parameters

- **text** (*str*) – Text of the button. If none of the optional fields are used, it will be sent as a message when the button is pressed.
- **request_contact** (*bool, optional*) – If True, the user's phone number will be sent as a contact when the button is pressed. Available in private chats only.
- **request_location** (*bool, optional*) – If True, the user's current location will be sent when the button is pressed. Available in private chats only.

class pyrogram.ReplyKeyboardRemove

Object used to tell clients to remove a bot keyboard.

Upon receiving a message with this object, Telegram clients will remove 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).

Parameters **selective** (*bool, optional*) – Use this parameter if you want to remove the 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 removes the keyboard for that user, while still showing the keyboard with poll options to users who haven't voted yet.

class pyrogram.InlineKeyboardMarkup

An inline keyboard that appears right next to the message it belongs to.

Parameters **inline_keyboard** (List of List of *InlineKeyboardButton*) – List of button rows, each represented by a List of *InlineKeyboardButton* objects.

class pyrogram.InlineKeyboardButton

One button of an inline keyboard.

You must use exactly one of the optional fields.

Parameters

- **text** (*str*) – Label text on the button.
- **callback_data** (*str|bytes, optional*) – Data to be sent in a callback query to the bot when button is pressed, 1-64 bytes.
- **url** (*str, optional*) – HTTP url to be opened when button is pressed.
- **switch_inline_query** (*str, optional*) – If set, pressing the button will prompt the user to select one of their chats, open that chat and insert the bot's username and the specified inline query in the input field. Can be empty, in which case just the bot's username will be inserted. Note: This offers an easy way for users to start using your bot in inline mode when they are currently in a private chat with it. Especially useful when combined with `switch_pm...` actions – in this case the user will be automatically returned to the chat they switched from, skipping the chat selection screen.
- **switch_inline_query_current_chat** (*str, optional*) – If set, pressing the button will insert the bot's username and the specified inline query in the current chat's input field.

Can be empty, in which case only the bot's username will be inserted. This offers a quick way for the user to open your bot in inline mode in the same chat – good for selecting something from multiple options.

class pyrogram.**ForceReply**

Object used to force clients to show a reply interface.

Upon receiving a message with this object, Telegram clients will display a reply interface to the user.

This acts 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.

Parameters **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.

class pyrogram.**CallbackQuery**

An incoming callback query from a callback button in an inline keyboard.

If the button that originated the query was attached to a message sent by the bot, the field *message* will be present. If the button was attached to a message sent via the bot (in inline mode), the field *inline_message_id* will be present. Exactly one of the fields *data* or *game_short_name* will be present.

Parameters

- **id** (*str*) – Unique identifier for this query.
- **from_user** (*User*) – Sender.
- **chat_instance** (*str, optional*) – Global identifier, uniquely corresponding to the chat to which the message with the callback button was sent. Useful for high scores in games.
- **message** (*Message, optional*) – Message with the callback button that originated the query. Note that message content and message date will not be available if the message is too old.
- **inline_message_id** (*str*) – Identifier of the message sent via the bot in inline mode, that originated the query.
- **data** (*str | bytes, optional*) – Data associated with the callback button. Be aware that a bad client can send arbitrary data in this field.
- **game_short_name** (*str, optional*) – Short name of a Game to be returned, serves as the unique identifier for the game.

class pyrogram.**GameHighScore**

One row of the high scores table for a game.

Parameters

- **user** (*User*) – User.
- **score** (*int*) – Score.
- **position** (*position, optional*) – Position in high score table for the game.

class pyrogram.**GameHighScores**

The high scores table for a game.

Parameters

- **total_count** (*int*) – Total number of scores the target game has.
- **game_high_scores** (List of *GameHighScore*) – Game scores.

class `pyrogram.CallbackGame`
Placeholder, currently holds no information.

Use `BotFather` to set up your game.

class `pyrogram.InputMedia`
Content of a media message to be sent.

It should be one of:

- `InputMediaAnimation`
- `InputMediaDocument`
- `InputMediaAudio`
- `InputMediaPhoto`
- `InputMediaVideo`

class `pyrogram.InputMediaPhoto`

A photo to be sent inside an album. It is intended to be used with `send_media_group()`.

Parameters

- **media** (`str`) – Photo to send. Pass a `file_id` as string to send a photo that exists on the Telegram servers or pass a file path as string to upload a new photo that exists on your local machine. Sending photo by a URL is currently unsupported.
- **caption** (`str`, *optional*) – Caption of the photo to be sent, 0-1024 characters
- **parse_mode** (`str`, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.

class `pyrogram.InputMediaVideo`

A video to be sent inside an album. It is intended to be used with `send_media_group()`.

Parameters

- **media** (`str`) – Video to send. Pass a `file_id` as string to send a video that exists on the Telegram servers or pass a file path as string to upload a new video that exists on your local machine. Sending video by a URL is currently unsupported.
- **thumb** (`str`) – Thumbnail of the video sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail’s width and height should not exceed 320 pixels. Thumbnails can’t be reused and can be only uploaded as a new file.
- **caption** (`str`, *optional*) – Caption of the video to be sent, 0-1024 characters
- **parse_mode** (`str`, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **width** (`int`, *optional*) – Video width.
- **height** (`int`, *optional*) – Video height.
- **duration** (`int`, *optional*) – Video duration.
- **supports_streaming** (`bool`, *optional*) – Pass True, if the uploaded video is suitable for streaming.

class `pyrogram.InputMediaAudio`

An audio to be sent inside an album.

It is intended to be used with `send_media_group()`.

Parameters

- **media** (*str*) – Audio to send. Pass a *file_id* as string to send an audio that exists on the Telegram servers or pass a file path as string to upload a new audio that exists on your local machine.
- **thumb** (*str, optional*) – Thumbnail of the music file album cover. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail's width and height should not exceed 320 pixels. Thumbnails can't be reused and can be only uploaded as a new file.
- **caption** (*str, optional*) – Caption of the audio to be sent, 0-1024 characters
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **duration** (*int, optional*) – Duration of the audio in seconds
- **performer** (*int, optional*) – Performer of the audio
- **title** (*int, optional*) – Title of the audio

class pyrogram.**InputMediaAnimation**

An animation file (GIF or H.264/MPEG-4 AVC video without sound) to be sent inside an album.

Parameters

- **media** (*str*) – Animation to send. Pass a *file_id* as string to send a file that exists on the Telegram servers or pass a file path as string to upload a new file that exists on your local machine.
- **thumb** (*str, optional*) – Thumbnail of the animation file sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail's width and height should not exceed 320 pixels. Thumbnails can't be reused and can be only uploaded as a new file.
- **caption** (*str, optional*) – Caption of the animation to be sent, 0-1024 characters
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **width** (*int, optional*) – Animation width.
- **height** (*int, optional*) – Animation height.
- **duration** (*int, optional*) – Animation duration.

class pyrogram.**InputMediaDocument**

A generic file to be sent inside an album.

Parameters

- **media** (*str*) – File to send. Pass a *file_id* as string to send a file that exists on the Telegram servers or pass a file path as string to upload a new file that exists on your local machine.
- **thumb** (*str*) – Thumbnail of the file sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail's width and height should not exceed 320 pixels. Thumbnails can't be reused and can be only uploaded as a new file.
- **caption** (*str, optional*) – Caption of the document to be sent, 0-1024 characters
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.

class pyrogram.**InputPhoneContact**

A Phone Contact to be added in your Telegram address book. It is intended to be used with [*add_contacts\(\)*](#)

Parameters

- **phone** (*str*) – Contact's phone number

- **first_name** (*str*) – Contact’s first name
- **last_name** (*str*, *optional*) – Contact’s last name

class pyrogram.**InlineQuery**

An incoming inline query.

When the user sends an empty query, your bot could return some default or trending results.

Parameters

- **id** (*str*) – Unique identifier for this query.
- **from_user** (*User*) – Sender.
- **query** (*str*) – Text of the query (up to 512 characters).
- **offset** (*str*) – Offset of the results to be returned, can be controlled by the bot.
- **location** (*Location*, *optional*) – Sender location, only for bots that request user location.

class pyrogram.**InlineQueryResult**

One result of an inline query.

Pyrogram currently supports results of the following 20 types:

- *InlineQueryResultArticle*

class pyrogram.**InlineQueryResultArticle**

Link to an article or web page.

TODO: Hide url?

Parameters

- **id** (*str*) – Unique identifier for this result, 1-64 bytes.
- **title** (*str*) – Title for the result.
- **input_message_content** (*InputMessageContent*) – Content of the message to be sent.
- **reply_markup** (*InlineKeyboardMarkup*, *optional*) – Inline keyboard attached to the message.
- **url** (*str*, *optional*) – URL of the result.
- **description** (*str*, *optional*) – Short description of the result.
- **thumb_url** (*str*, *optional*) – Url of the thumbnail for the result.
- **thumb_width** (*int*, *optional*) – Thumbnail width.
- **thumb_height** (*int*, *optional*) – Thumbnail height.

class pyrogram.**InputMessageContent**

Content of a message to be sent as a result of an inline query.

Pyrogram currently supports the following 4 types:

- *InputTextMessageContent*

class pyrogram.**InputTextMessageContent**

Content of a text message to be sent as the result of an inline query.

Parameters

- **message_text** (*str*) – Text of the message to be sent, 1-4096 characters.

- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your message. Defaults to “markdown”.
- **disable_web_page_preview** (*bool, optional*) – Disables link previews for links in this message.

4.11 Bound Methods

Some Pyrogram types define what are called bound methods. Bound methods are functions attached to a class which are accessed via an instance of that class. They make it even easier to call specific methods by automatically inferring some of the required arguments.

```
from pyrogram import Client

app = Client("my_account")

@app.on_message()
def hello(client, message)
    message.reply("hi")

app.run()
```

- *Message*
- *CallbackQuery*
- *InlineQuery*

4.11.1 Message

- *Message.click()*
- *Message.delete()*
- *Message.download()*
- *Message.edit()*
- *Message.edit_caption()*
- *Message.edit_media()*
- *Message.edit_reply_markup()*
- *Message.forward()*
- *Message.pin()*
- *Message.reply()*
- *Message.reply_animation()*
- *Message.reply_audio()*
- *Message.reply_cached_media()*
- *Message.reply_chat_action()*
- *Message.reply_contact()*

- `Message.reply_document()`
- `Message.reply_game()`
- `Message.reply_inline_bot_result()`
- `Message.reply_location()`
- `Message.reply_media_group()`
- `Message.reply_photo()`
- `Message.reply_poll()`
- `Message.reply_sticker()`
- `Message.reply_venue()`
- `Message.reply_video()`
- `Message.reply_video_note()`
- `Message.reply_voice()`

`Message.click()`

Bound method *click* `Message`.

Use as a shortcut for clicking a button attached to the message instead of:

- Clicking inline buttons:

```
client.request_callback_answer(
    chat_id=message.chat.id,
    message_id=message.message_id,
    callback_data=message.reply_markup[i][j].callback_data
)
```

- Clicking normal buttons:

```
client.send_message(
    chat_id=message.chat.id,
    text=message.reply_markup[i][j].text
)
```

Example

This method can be used in three different ways:

1. Pass one integer argument only (e.g.: `.click(2)`), to click a button at index 2). Buttons are counted left to right, starting from the top.
2. Pass two integer arguments (e.g.: `.click(1, 0)`), to click a button at position (1, 0)). The origin (0, 0) is top-left.
3. Pass one string argument only (e.g.: `.click("Settings")`), to click a button by using its label). Only the first matching button will be pressed.

Parameters

- **x** (`int` | `str`) – Used as integer index, integer abscissa (in pair with y) or as string label. Defaults to 0 (first button).

- **y** (*int, optional*) – Used as ordinate only (in pair with x).
- **quote** (*bool, optional*) – Useful for normal buttons only, where pressing it will result in a new message sent. If `True`, the message will be sent as a reply to this message. Defaults to `True` in group chats and `False` in private chats.
- **timeout** (*int, optional*) – Timeout in seconds.

Returns

- The result of `request_callback_answer()` in case of inline callback button clicks.
- The result of `reply()` in case of normal button clicks.
- A string in case the inline button is a URL, a `switch_inline_query` or a `switch_inline_query_current_chat` button.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` – In case the provided index or position is out of range or the button label was not found.
- `TimeoutError` – In case, after clicking an inline button, the bot fails to answer within the timeout.

`Message.delete()`

Bound method `delete` *Message*.

Use as a shortcut for:

```
client.delete_messages(  
    chat_id=chat_id,  
    message_ids=message.message_id  
)
```

Example

```
message.delete()
```

Parameters **revoke** (*bool, optional*) – Deletes messages on both parts. This is only for private cloud chats and normal groups, messages on channels and supergroups are always revoked (i.e.: deleted for everyone). Defaults to `True`.

Returns `True` on success, `False` otherwise.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.download()`

Bound method `download` *Message*.

Use as a shortcut for:

```
client.download_media(message)
```


Example

```
message.download()
```

Parameters

- **file_name** (*str, optional*) – A custom *file_name* to be used instead of the one provided by Telegram. By default, all files are downloaded in the *downloads* folder in your working directory. You can also specify a path for downloading files in a custom location: paths that end with “/” are considered directories. All non-existent folders will be created automatically.
- **block** (*bool, optional*) – Blocks the code execution until the file has been downloaded. Defaults to True.
- **progress** (*callable*) – Pass a callback function to view the download progress. The function must take (*client, current, total, *args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Returns On success, the absolute path of the downloaded file as string is returned, None otherwise.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` – If the message doesn’t contain any downloadable media

`Message.edit()`

Bound method *edit* *Message*.

Use as a shortcut for:

```
client.edit_message_text(
    chat_id=message.chat.id,
    message_id=message.message_id,
    text="hello"
)
```

Example

```
message.edit("hello")
```

Parameters

- **text** (*str*) – New text of the message.
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your message. Defaults to “markdown”.
- **disable_web_page_preview** (*bool, optional*) – Disables link previews for links in this message.
- **reply_markup** (*InlineKeyboardMarkup, optional*) – An *InlineKeyboardMarkup* object.

Returns On success, the edited *Message* is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Message.edit_caption()`

Bound method *edit_caption* *Message*.

Use as a shortcut for:

```
client.edit_message_caption(
    chat_id=message.chat.id,
    message_id=message.message_id,
    caption="hello"
)
```

Example

```
message.edit_caption("hello")
```

Parameters

- **caption** (*str*) – New caption of the message.
- **parse_mode** (*str*, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your message. Defaults to “markdown”.
- **reply_markup** (*InlineKeyboardMarkup*, *optional*) – An *InlineKeyboardMarkup* object.

Returns On success, the edited *Message* is returned.

Raises *RPCError* – In case of a Telegram RPC error.

`Message.edit_media()`

Bound method *edit_media* *Message*.

Use as a shortcut for:

```
client.edit_message_media(
    chat_id=message.chat.id,
    message_id=message.message_id,
    media=media
)
```

Example

```
message.edit_media(media)
```

Parameters

- **media** (*InputMediaAnimation* | *InputMediaAudio* | *InputMediaDocument* | *InputMediaPhoto* | *InputMediaVideo*) – One of the *InputMedia* objects describing an animation, audio, document, photo or video.
- **reply_markup** (*InlineKeyboardMarkup*, *optional*) – An *InlineKeyboardMarkup* object.

Returns On success, the edited *Message* is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.edit_reply_markup()`

Bound method *edit_reply_markup* *Message*.

Use as a shortcut for:

```
client.edit_message_reply_markup(
    chat_id=message.chat.id,
    message_id=message.message_id,
    reply_markup=inline_reply_markup
)
```

Example

```
message.edit_reply_markup(inline_reply_markup)
```

Parameters `reply_markup` (*InlineKeyboardMarkup*) – An *InlineKeyboardMarkup* object.

Returns On success, if edited message is sent by the bot, the edited *Message* is returned, otherwise `True` is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.forward()`

Bound method *forward* *Message*.

Use as a shortcut for:

```
client.forward_messages(
    chat_id=chat_id,
    from_chat_id=message.chat.id,
    message_ids=message.message_id
)
```

Example

```
message.forward(chat_id)
```

Parameters

- **chat_id** (`int` | `str`) – Unique identifier (`int`) or username (`str`) of the target chat. For your personal cloud (Saved Messages) you can simply use “me” or “self”. For a contact that exists in your Telegram address book you can use his phone number (`str`).
- **disable_notification** (`bool`, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **as_copy** (`bool`, *optional*) – Pass `True` to forward messages without the forward header (i.e.: send a copy of the message content). Defaults to `False`.
- **remove_caption** (`bool`, *optional*) – If set to `True` and *as_copy* is enabled as well, media captions are not preserved when copying the message. Has no effect if *as_copy* is not enabled. Defaults to `False`.

Returns On success, the forwarded Message is returned.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.pin()`

Bound method *pin* *Message*.

Use as a shortcut for:

```
client.pin_chat_message(
    chat_id=message.chat.id,
    message_id=message_id
)
```

Example

```
message.pin()
```

Parameters `disable_notification` (`bool`) – Pass `True`, if it is not necessary to send a notification to all chat members about the new pinned message. Notifications are always disabled in channels.

Returns `True` on success.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.reply()`

Bound method *reply* *Message*.

Use as a shortcut for:

```
client.send_message(
    chat_id=message.chat.id,
    text="hello",
    reply_to_message_id=message.message_id
)
```

Example

```
message.reply("hello", quote=True)
```

Parameters

- **text** (`str`) – Text of the message to be sent.
- **quote** (`bool`, *optional*) – If `True`, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to `True` in group chats and `False` in private chats.
- **parse_mode** (`str`, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your message. Defaults to “markdown”.
- **disable_web_page_preview** (`bool`, *optional*) – Disables link previews for links in this message.

- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

Returns On success, the sent Message is returned.

Raises RPCError – In case of a Telegram RPC error.

Message.**reply_animation**()

Bound method *reply_animation* Message.

Use as a shortcut for:

```
client.send_animation(
    chat_id=message.chat.id,
    animation=animation
)
```

Example

```
message.reply_animation(animation)
```

Parameters

- **animation** (*str*) – Animation to send. Pass a file_id as string to send an animation that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get an animation from the Internet, or pass a file path as string to upload a new animation that exists on your local machine.
- **quote** (*bool, optional*) – If True, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to True in group chats and False in private chats.
- **caption** (*str, optional*) – Animation caption, 0-1024 characters.
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **duration** (*int, optional*) – Duration of sent animation in seconds.
- **width** (*int, optional*) – Animation width.
- **height** (*int, optional*) – Animation height.
- **thumb** (*str, optional*) – Thumbnail of the animation file sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail’s width and height should not exceed 320 pixels. Thumbnails can’t be reused and can be only uploaded as a new file.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.

- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable*, *optional*) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple*, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple*, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns On success, the sent *Message* is returned. In case the upload is deliberately stopped with `stop_transmission()`, `None` is returned instead.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.reply_audio()`

Bound method *reply_audio* *Message*.

Use as a shortcut for:

```
client.send_audio(
    chat_id=message.chat.id,
    audio=audio
)
```

Example

```
message.reply_audio(audio)
```

Parameters

- **audio** (*str*) – Audio file to send. Pass a *file_id* as string to send an audio file that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get an audio file from the Internet, or pass a file path as string to upload a new audio file that exists on your local machine.
- **quote** (*bool*, *optional*) – If `True`, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to `True` in group chats and `False` in private chats.
- **caption** (*str*, *optional*) – Audio caption, 0-1024 characters.
- **parse_mode** (*str*, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.

- **duration** (*int, optional*) – Duration of the audio in seconds.
- **performer** (*str, optional*) – Performer.
- **title** (*str, optional*) – Track name.
- **thumb** (*str, optional*) – Thumbnail of the music file album cover. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail's width and height should not exceed 320 pixels. Thumbnails can't be reused and can be only uploaded as a new file.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable, optional*) – Pass a callback function to view the upload progress. The function must take (*client, current, total, *args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple, optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple, optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns On success, the sent *Message* is returned. In case the upload is deliberately stopped with *stop_transmission()*, *None* is returned instead.

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_cached_media()`
Bound method *reply_cached_media* *Message*.

Use as a shortcut for:

```
client.send_cached_media(
    chat_id=message.chat.id,
    file_id=file_id
)
```

Example

```
message.reply_cached_media(file_id)
```

Parameters

- **file_id** (*str*) – Media to send. Pass a `file_id` as string to send a media that exists on the Telegram servers.
- **quote** (*bool, optional*) – If `True`, the message will be sent as a reply to this message. If `reply_to_message_id` is passed, this parameter will be ignored. Defaults to `True` in group chats and `False` in private chats.
- **caption** (*bool, optional*) – Media caption, 0-1024 characters.
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

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

Raises `RPCError` – In case of a Telegram RPC error.

`Message.reply_chat_action()`

Bound method `reply_chat_action` *Message*.

Use as a shortcut for:

```
client.send_chat_action(
    chat_id=message.chat.id,
    action="typing"
)
```

Example

```
message.reply_chat_action("typing")
```

Parameters **action** (*str*) – 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, “record_video_note” or “upload_video_note” for video notes, “choose_contact” for contacts, “playing” for games or “cancel” to cancel any chat action currently displayed.

Returns `bool` – On success, `True` is returned.

Raises

- `RPCError` – In case of a Telegram RPC error.
- `ValueError` – In case the provided string is not a valid chat action.

`Message.reply_contact()`

Bound method `reply_contact` *Message*.

Use as a shortcut for:


```
client.send_contact(
    chat_id=message.chat.id,
    phone_number=phone_number,
    first_name=first_name
)
```

Example

```
message.reply_contact(phone_number, "Dan")
```

Parameters

- **phone_number** (*str*) – Contact’s phone number.
- **first_name** (*str*) – Contact’s first name.
- **quote** (*bool, optional*) – If *True*, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to *True* in group chats and *False* in private chats.
- **last_name** (*str, optional*) – Contact’s last name.
- **vcard** (*str, optional*) – Additional data about the contact in the form of a vCard, 0-2048 bytes
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

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

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_document()`

Bound method *reply_document* *Message*.

Use as a shortcut for:

```
client.send_document(
    chat_id=message.chat.id,
    document=document
)
```

Example

```
message.reply_document(document)
```

Parameters

- **document** (*str*) – File to send. Pass a *file_id* as string to send a file that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get a file from the Internet, or pass a file path as string to upload a new file that exists on your local machine.
- **quote** (*bool, optional*) – If *True*, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to *True* in group chats and *False* in private chats.
- **thumb** (*str, optional*) – Thumbnail of the file sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail's width and height should not exceed 320 pixels. Thumbnails can't be reused and can be only uploaded as a new file.
- **caption** (*str, optional*) – Document caption, 0-1024 characters.
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable, optional*) – Pass a callback function to view the upload progress. The function must take (*client, current, total, *args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple, optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple, optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns On success, the sent *Message* is returned. In case the upload is deliberately stopped with *stop_transmission()*, *None* is returned instead.

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_game()`

Bound method *reply_game* *Message*.

Use as a shortcut for:

```
client.send_game(  
    chat_id=message.chat.id,  
    game_short_name="lumberjack"  
)
```

Example

```
message.reply_game("lumberjack")
```

Parameters

- **game_short_name** (*str*) – Short name of the game, serves as the unique identifier for the game. Set up your games via Botfather.
- **quote** (*bool, optional*) – If `True`, the message will be sent as a reply to this message. If `reply_to_message_id` is passed, this parameter will be ignored. Defaults to `True` in group chats and `False` in private chats.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup, optional*) – An object for an inline keyboard. If empty, one ‘Play game_title’ button will be shown automatically. If not empty, the first button must launch the game.

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

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_inline_bot_result()`

Bound method *reply_inline_bot_result* *Message*.

Use as a shortcut for:

```
client.send_inline_bot_result(
    chat_id=message.chat.id,
    query_id=query_id,
    result_id=result_id
)
```

Example

```
message.reply_inline_bot_result(query_id, result_id)
```

Parameters

- **query_id** (*int*) – Unique identifier for the answered query.
- **result_id** (*str*) – Unique identifier for the result that was chosen.
- **quote** (*bool, optional*) – If `True`, the message will be sent as a reply to this message. If `reply_to_message_id` is passed, this parameter will be ignored. Defaults to `True` in group chats and `False` in private chats.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*bool, optional*) – If the message is a reply, ID of the original message.
- **hide_via** (*bool*) – Sends the message with *via @bot* hidden.

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

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_location()`

Bound method *reply_location* *Message*.

Use as a shortcut for:

```
client.send_location(
    chat_id=message.chat.id,
    latitude=41.890251,
    longitude=12.492373
)
```

Example

```
message.reply_location(41.890251, 12.492373)
```

Parameters

- **latitude** (float) – Latitude of the location.
- **longitude** (float) – Longitude of the location.
- **quote** (bool, *optional*) – If True, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to True in group chats and False in private chats.
- **disable_notification** (bool, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (int, *optional*) – If the message is a reply, ID of the original message
- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

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

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_media_group()`

Bound method *reply_media_group* *Message*.

Use as a shortcut for:

```
client.send_media_group(
    chat_id=message.chat.id,
    media=list_of_media
)
```

Example

```
message.reply_media_group(list_of_media)
```

Parameters

- **media** (*list*) – A list containing either *InputMediaPhoto* or *InputMediaVideo* objects describing photos and videos to be sent, must include 2–10 items.
- **quote** (*bool, optional*) – If *True*, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to *True* in group chats and *False* in private chats.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.

Returns On success, a *Messages* object is returned containing all the single messages sent.

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_photo()`

Bound method *reply_photo* *Message*.

Use as a shortcut for:

```
client.send_photo(
    chat_id=message.chat.id,
    photo=photo
)
```

Example

```
message.reply_photo(photo)
```

Parameters

- **photo** (*str*) – Photo to send. Pass a *file_id* as string to send a photo that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get a photo from the Internet, or pass a file path as string to upload a new photo that exists on your local machine.
- **quote** (*bool, optional*) – If *True*, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to *True* in group chats and *False* in private chats.
- **caption** (*bool, optional*) – Photo caption, 0-1024 characters.
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **ttl_seconds** (*int, optional*) – Self-Destruct Timer. If you set a timer, the photo will self-destruct in *ttl_seconds* seconds after it was viewed.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.

- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable, optional*) – Pass a callback function to view the upload progress. The function must take (*client, current, total, *args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple, optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple, optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns On success, the sent *Message* is returned. In case the upload is deliberately stopped with *stop_transmission()*, *None* is returned instead.

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_poll()`

Bound method *reply_poll* *Message*.

Use as a shortcut for:

```
client.send_poll(
    chat_id=message.chat.id,
    question="Is Pyrogram the best?",
    options=["Yes", "Yes"]
)
```

Example

```
message.reply_poll("Is Pyrogram the best?", ["Yes", "Yes"])
```

Parameters

- **question** (*str*) – The poll question, as string.
- **options** (*List of str*) – The poll options, as list of strings (2 to 10 options are allowed).
- **quote** (*bool, optional*) – If *True*, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to *True* in group chats and *False* in private chats.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.

- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

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

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_sticker()`

Bound method *reply_sticker* *Message*.

Use as a shortcut for:

```
client.send_sticker(
    chat_id=message.chat.id,
    sticker=sticker
)
```

Example

```
message.reply_sticker(sticker)
```

Parameters

- **sticker** (*str*) – Sticker to send. Pass a *file_id* as string to send a sticker that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get a .webp sticker file from the Internet, or pass a file path as string to upload a new sticker that exists on your local machine.
- **quote** (*bool, optional*) – If *True*, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to *True* in group chats and *False* in private chats.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message.
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable, optional*) – Pass a callback function to view the upload progress. The function must take (*client, current, total, *args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple, optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (int) – The amount of bytes uploaded so far.
- **total** (int) – The size of the file.
- ***args** (tuple, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns On success, the sent *Message* is returned. In case the upload is deliberately stopped with `stop_transmission()`, `None` is returned instead.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.reply_venue()`

Bound method *reply_venue* *Message*.

Use as a shortcut for:

```
client.send_venue(
    chat_id=message.chat.id,
    latitude=41.890251,
    longitude=12.492373,
    title="Coliseum",
    address="Piazza del Colosseo, 1, 00184 Roma RM"
)
```

Example

```
message.reply_venue(41.890251, 12.492373, "Coliseum", "Piazza del Colosseo, 1, ↵
↪00184 Roma RM")
```

Parameters

- **latitude** (float) – Latitude of the venue.
- **longitude** (float) – Longitude of the venue.
- **title** (str) – Name of the venue.
- **address** (str) – Address of the venue.
- **quote** (bool, *optional*) – If `True`, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to `True` in group chats and `False` in private chats.
- **foursquare_id** (str, *optional*) – Foursquare identifier of the venue.
- **foursquare_type** (str, *optional*) – Foursquare type of the venue, if known. (For example, “arts_entertainment/default”, “arts_entertainment/aquarium” or “food/icecream”.)
- **disable_notification** (bool, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (int, *optional*) – If the message is a reply, ID of the original message

- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.

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

Raises *RPCError* – In case of a Telegram RPC error.

`Message.reply_video()`

Bound method *reply_video* *Message*.

Use as a shortcut for:

```
client.send_video(
    chat_id=message.chat.id,
    video=video
)
```

Example

```
message.reply_video(video)
```

Parameters

- **video** (*str*) – Video to send. Pass a *file_id* as string to send a video that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get a video from the Internet, or pass a file path as string to upload a new video that exists on your local machine.
- **quote** (*bool*, *optional*) – If *True*, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to *True* in group chats and *False* in private chats.
- **caption** (*str*, *optional*) – Video caption, 0-1024 characters.
- **parse_mode** (*str*, *optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **duration** (*int*, *optional*) – Duration of sent video in seconds.
- **width** (*int*, *optional*) – Video width.
- **height** (*int*, *optional*) – Video height.
- **thumb** (*str*, *optional*) – Thumbnail of the video sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail’s width and height should not exceed 320 pixels. Thumbnails can’t be reused and can be only uploaded as a new file.
- **supports_streaming** (*bool*, *optional*) – Pass *True*, if the uploaded video is suitable for streaming.
- **disable_notification** (*bool*, *optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int*, *optional*) – If the message is a reply, ID of the original message.

- **reply_markup** (*InlineKeyboardMarkup* | *ReplyKeyboardMarkup* | *ReplyKeyboardRemove* | *ForceReply*, *optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable*, *optional*) – Pass a callback function to view the upload progress. The function must take (*client*, *current*, *total*, **args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple*, *optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple*, *optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns On success, the sent *Message* is returned. In case the upload is deliberately stopped with `stop_transmission()`, `None` is returned instead.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.reply_video_note()`

Bound method *reply_video_note* *Message*.

Use as a shortcut for:

```
client.send_video_note(
    chat_id=message.chat.id,
    video_note=video_note
)
```

Example

```
message.reply_video_note(video_note)
```

Parameters

- **video_note** (*str*) – Video note to send. Pass a *file_id* as string to send a video note that exists on the Telegram servers, or pass a file path as string to upload a new video note that exists on your local machine. Sending video notes by a URL is currently unsupported.
- **quote** (*bool*, *optional*) – If `True`, the message will be sent as a reply to this message. If *reply_to_message_id* is passed, this parameter will be ignored. Defaults to `True` in group chats and `False` in private chats.
- **duration** (*int*, *optional*) – Duration of sent video in seconds.
- **length** (*int*, *optional*) – Video width and height.

- **thumb** (*str, optional*) – Thumbnail of the video sent. The thumbnail should be in JPEG format and less than 200 KB in size. A thumbnail's width and height should not exceed 320 pixels. Thumbnails can't be reused and can be only uploaded as a new file.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message
- **reply_markup** (*InlineKeyboardMarkup | ReplyKeyboardMarkup | ReplyKeyboardRemove | ForceReply, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable, optional*) – Pass a callback function to view the upload progress. The function must take (*client, current, total, *args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple, optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a *chat_id* and a *message_id* in order to edit a message with the updated progress.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple, optional*) – Extra custom arguments as defined in the *progress_args* parameter. You can either keep **args* or add every single extra argument in your function signature.

Returns On success, the sent *Message* is returned. In case the upload is deliberately stopped with `stop_transmission()`, `None` is returned instead.

Raises `RPCError` – In case of a Telegram RPC error.

`Message.reply_voice()`

Bound method *reply_voice* *Message*.

Use as a shortcut for:

```
client.send_voice(
    chat_id=message.chat.id,
    voice=voice
)
```

Example

```
message.reply_voice(voice)
```

Parameters

- **voice** (*str*) – Audio file to send. Pass a *file_id* as string to send an audio that exists on the Telegram servers, pass an HTTP URL as a string for Telegram to get an audio from the Internet, or pass a file path as string to upload a new audio that exists on your local machine.

- **quote** (*bool, optional*) – If `True`, the message will be sent as a reply to this message. If `reply_to_message_id` is passed, this parameter will be ignored. Defaults to `True` in group chats and `False` in private chats.
- **caption** (*str, optional*) – Voice message caption, 0-1024 characters.
- **parse_mode** (*str, optional*) – Pass “markdown” or “html” if you want Telegram apps to show bold, italic, fixed-width text or inline URLs in your caption. Defaults to “markdown”.
- **duration** (*int, optional*) – Duration of the voice message in seconds.
- **disable_notification** (*bool, optional*) – Sends the message silently. Users will receive a notification with no sound.
- **reply_to_message_id** (*int, optional*) – If the message is a reply, ID of the original message
- **reply_markup** (*`InlineKeyboardMarkup` | `ReplyKeyboardMarkup` | `ReplyKeyboardRemove` | `ForceReply`, optional*) – Additional interface options. An object for an inline keyboard, custom reply keyboard, instructions to remove reply keyboard or to force a reply from the user.
- **progress** (*callable, optional*) – Pass a callback function to view the upload progress. The function must take (*client, current, total, *args*) as positional arguments (look at the section below for a detailed description).
- **progress_args** (*tuple, optional*) – Extra custom arguments for the progress callback function. Useful, for example, if you want to pass a `chat_id` and a `message_id` in order to edit a message with the updated progress.

Other Parameters

- **client** (*`Client`*) – The Client itself, useful when you want to call other API methods inside the callback function.
- **current** (*int*) – The amount of bytes uploaded so far.
- **total** (*int*) – The size of the file.
- ***args** (*tuple, optional*) – Extra custom arguments as defined in the `progress_args` parameter. You can either keep `*args` or add every single extra argument in your function signature.

Returns On success, the sent `Message` is returned. In case the upload is deliberately stopped with `stop_transmission()`, `None` is returned instead.

Raises `RPCError` – In case of a Telegram RPC error.

4.11.2 CallbackQuery

`CallbackQuery.answer()`

Bound method `answer` of `CallbackQuery`.

Use this method as a shortcut for:

```
client.answer_callback_query(  
    callback_query.id,  
    text="Hello",  
    show_alert=True  
)
```

Example

```
callback_query.answer("Hello", show_alert=True)
```

Parameters

- **text** (*str*) – Text of the notification. If not specified, nothing will be shown to the user, 0-200 characters.
- **show_alert** (*bool*) – If true, an alert will be shown by the client instead of a notification at the top of the chat screen. Defaults to False.
- **url** (*str*) – URL that will be opened by the user's client. If you have created a Game and accepted the conditions via @Botfather, specify the URL that opens your game – note that this will only work if the query comes from a callback_game button. Otherwise, you may use links like t.me/your_bot?start=XXXX that open your bot with a parameter.
- **cache_time** (*int*) – The maximum amount of time in seconds that the result of the callback query may be cached client-side. Telegram apps will support caching starting in version 3.14. Defaults to 0.

4.11.3 InlineQuery

`InlineQuery.answer()`

Bound method *answer* of *InlineQuery*.

Use this method as a shortcut for:

```
client.answer_inline_query(
    inline_query.id,
    results=[...]
)
```

Example

```
inline_query.answer([...])
```

Parameters

- **results** (List of *InlineQueryResult*) – A list of results for the inline query.
- **cache_time** (*int, optional*) – The maximum amount of time in seconds that the result of the inline query may be cached on the server. Defaults to 300.
- **is_personal** (*bool, optional*) – Pass True, if results may be cached on the server side only for the user that sent the query. By default, results may be returned to any user who sends the same query.
- **next_offset** (*str, optional*) – Pass the offset that a client should send in the next query with the same text to receive more results. Pass an empty string if there are no more results or if you don't support pagination. Offset length can't exceed 64 bytes.
- **switch_pm_text** (*str, optional*) – If passed, clients will display a button with specified text that switches the user to a private chat with the bot and sends the bot a start message with the parameter `switch_pm_parameter`

- **switch_pm_parameter** (*str, optional*) – Deep-linking parameter for the /start message sent to the bot when user presses the switch button. 1-64 characters, only A-Z, a-z, 0-9, _ and - are allowed.

Example: An inline bot that sends YouTube videos can ask the user to connect the bot to their YouTube account to adapt search results accordingly. To do this, it displays a “Connect your YouTube account” button above the results, or even before showing any. The user presses the button, switches to a private chat with the bot and, in doing so, passes a start parameter that instructs the bot to return an oauth link. Once done, the bot can offer a switch_inline button so that the user can easily return to the chat where they wanted to use the bot’s inline capabilities.

4.12 Update Handlers

Handlers are used to instruct Pyrogram about which kind of updates you’d like to handle with your callback functions.

For a much more convenient way of registering callback functions have a look at [Decorators](#) instead. In case you decided to manually create a handler, use `add_handler()` to register it.

```
from pyrogram import Client, MessageHandler

app = Client("my_account")

def dump(client, message):
    print(message)

app.add_handler(MessageHandler(dump))

app.run()
```

<code>MessageHandler</code>	The Message handler class.
<code>DeletedMessagesHandler</code>	The deleted Messages handler class.
<code>CallbackQueryHandler</code>	The CallbackQuery handler class.
<code>InlineQueryHandler</code>	The InlineQuery handler class.
<code>UserStatusHandler</code>	The UserStatus handler class.
<code>PollHandler</code>	The Poll handler class.
<code>DisconnectHandler</code>	The Disconnect handler class.
<code>RawUpdateHandler</code>	The Raw Update handler class.

class pyrogram.MessageHandler

The Message handler class. Used to handle text, media and service messages coming from any chat (private, group, channel). It is intended to be used with `add_handler()`

For a nicer way to register this handler, have a look at the `on_message()` decorator.

Parameters

- **callback** (*callable*) – Pass a function that will be called when a new Message arrives. It takes (*client, message*) as positional arguments (look at the section below for a detailed description).
- **filters** (*Filters*) – Pass one or more filters to allow only a subset of messages to be passed in your callback function.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the message handler.
- **message** (*Message*) – The received message.

class pyrogram.DeletedMessagesHandler

The deleted Messages handler class. Used to handle deleted messages coming from any chat (private, group, channel). It is intended to be used with `add_handler()`

For a nicer way to register this handler, have a look at the `on_deleted_messages()` decorator.

Parameters

- **callback** (callable) – Pass a function that will be called when one or more Messages have been deleted. It takes (*client, messages*) as positional arguments (look at the section below for a detailed description).
- **filters** (*Filters*) – Pass one or more filters to allow only a subset of messages to be passed in your callback function.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the message handler.
- **messages** (*Messages*) – The deleted messages.

class pyrogram.CallbackQueryHandler

The CallbackQuery handler class. Used to handle callback queries coming from inline buttons. It is intended to be used with `add_handler()`

For a nicer way to register this handler, have a look at the `on_callback_query()` decorator.

Parameters

- **callback** (callable) – Pass a function that will be called when a new CallbackQuery arrives. It takes (*client, callback_query*) as positional arguments (look at the section below for a detailed description).
- **filters** (*Filters*) – Pass one or more filters to allow only a subset of callback queries to be passed in your callback function.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the message handler.
- **callback_query** (*CallbackQuery*) – The received callback query.

class pyrogram.InlineQueryHandler

The InlineQuery handler class. Used to handle inline queries. It is intended to be used with `add_handler()`

For a nicer way to register this handler, have a look at the `on_inline_query()` decorator.

Parameters

- **callback** (callable) – Pass a function that will be called when a new InlineQuery arrives. It takes (*client, inline_query*) as positional arguments (look at the section below for a detailed description).
- **filters** (*Filters*) – Pass one or more filters to allow only a subset of inline queries to be passed in your callback function.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the inline query handler.
- **inline_query** (*InlineQuery*) – The received inline query.

class pyrogram.**UserStatusHandler**

The UserStatus handler class. Used to handle user status updates (user going online or offline). It is intended to be used with `add_handler()`

For a nicer way to register this handler, have a look at the `on_user_status()` decorator.

Parameters

- **callback** (callable) – Pass a function that will be called when a new UserStatus update arrives. It takes (*client*, *user_status*) as positional arguments (look at the section below for a detailed description).
- **filters** (*Filters*) – Pass one or more filters to allow only a subset of messages to be passed in your callback function.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the user status handler.
- **user_status** (*UserStatus*) – The received UserStatus update.

class pyrogram.**PollHandler**

The Poll handler class. Used to handle polls updates.

It is intended to be used with `add_handler()`

For a nicer way to register this handler, have a look at the `on_poll()` decorator.

Parameters

- **callback** (callable) – Pass a function that will be called when a new poll update arrives. It takes (*client*, *poll*) as positional arguments (look at the section below for a detailed description).
- **filters** (*Filters*) – Pass one or more filters to allow only a subset of polls to be passed in your callback function.

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the poll handler.
- **poll** (*Poll*) – The received poll.

class pyrogram.**DisconnectHandler**

The Disconnect handler class. Used to handle disconnections. It is intended to be used with `add_handler()`

For a nicer way to register this handler, have a look at the `on_disconnect()` decorator.

Parameters **callback** (callable) – Pass a function that will be called when a disconnection occurs. It takes (*client*) as positional argument (look at the section below for a detailed description).

Other Parameters **client** (*Client*) – The Client itself. Useful, for example, when you want to change the proxy before a new connection is established.

class pyrogram.**RawUpdateHandler**

The Raw Update handler class. Used to handle raw updates. It is intended to be used with `add_handler()`

For a nicer way to register this handler, have a look at the `on_raw_update()` decorator.

Parameters `callback` (callable) – A function that will be called when a new update is received from the server. It takes (*client*, *update*, *users*, *chats*) as positional arguments (look at the section below for a detailed description).

Other Parameters

- **client** (*Client*) – The Client itself, useful when you want to call other API methods inside the update handler.
- **update** (*Update*) – The received update, which can be one of the many single Updates listed in the *updates* field you see in the *Update* type.
- **users** (dict) – Dictionary of all *User* mentioned in the update. You can access extra info about the user (such as *first_name*, *last_name*, etc...) by using the IDs you find in the *update* argument (e.g.: *users[1768841572]*).
- **chats** (dict) – Dictionary of all *Chat* and *Channel* mentioned in the update. You can access extra info about the chat (such as *title*, *participants_count*, etc...) by using the IDs you find in the *update* argument (e.g.: *chats[1701277281]*).

Note: The following Empty or Forbidden types may exist inside the *users* and *chats* dictionaries. They mean you have been blocked by the user or banned from the group/channel.

- *UserEmpty*
 - *ChatEmpty*
 - *ChatForbidden*
 - *ChannelForbidden*
-

4.13 Decorators

While still being methods bound to the *Client* class, decorators are of a special kind and thus deserve a dedicated page.

Decorators are able to register callback functions for handling updates in a much easier and cleaner way compared to *Handlers*; they do so by instantiating the correct handler and calling *add_handler()*, automatically. All you need to do is adding the decorators on top of your functions.

```
from pyrogram import Client

app = Client("my_account")

@app.on_message()
def log(client, message):
    print(message)

app.run()
```

<i>on_message</i>	Decorator for handling messages.
<i>on_callback_query</i>	Decorator for handling callback queries.
<i>on_inline_query</i>	Decorator for handling inline queries.

Continued on next page

Table 15 – continued from previous page

<code>on_deleted_messages</code>	Decorator for handling deleted messages.
<code>on_user_status</code>	Decorator for handling user status updates.
<code>on_poll</code>	Decorator for handling poll updates.
<code>on_disconnect</code>	Decorator for handling disconnections.
<code>on_raw_update</code>	Decorator for handling raw updates.

`Client.on_message()`

Decorator for handling messages.

This does the same thing as `add_handler()` using the `MessageHandler`.

Parameters

- **filters** (`Filters`) – Pass one or more filters to allow only a subset of messages to be passed in your function.
- **group** (`int`, *optional*) – The group identifier, defaults to 0.

`Client.on_callback_query()`

Decorator for handling callback queries.

This does the same thing as `add_handler()` using the `CallbackQueryHandler`.

Parameters

- **filters** (`Filters`) – Pass one or more filters to allow only a subset of callback queries to be passed in your function.
- **group** (`int`, *optional*) – The group identifier, defaults to 0.

`Client.on_inline_query()`

Decorator for handling inline queries.

This does the same thing as `add_handler()` using the `InlineQueryHandler`.

Parameters

- **filters** (`Filters`) – Pass one or more filters to allow only a subset of inline queries to be passed in your function.
- **group** (`int`, *optional*) – The group identifier, defaults to 0.

`Client.on_deleted_messages()`

Decorator for handling deleted messages.

This does the same thing as `add_handler()` using the `DeletedMessagesHandler`.

Parameters

- **filters** (`Filters`) – Pass one or more filters to allow only a subset of messages to be passed in your function.
- **group** (`int`, *optional*) – The group identifier, defaults to 0.

`Client.on_user_status()`

Decorator for handling user status updates. This does the same thing as `add_handler()` using the `UserStatusHandler`.

Parameters

- **filters** (`Filters`) – Pass one or more filters to allow only a subset of UserStatus updated to be passed in your function.
- **group** (`int`, *optional*) – The group identifier, defaults to 0.

`Client.on_poll()`

Decorator for handling poll updates.

This does the same thing as `add_handler()` using the `PollHandler`.

Parameters

- **filters** (`Filters`) – Pass one or more filters to allow only a subset of polls to be passed in your function.
- **group** (`int`, *optional*) – The group identifier, defaults to 0.

`Client.on_disconnect()`

Decorator for handling disconnections.

This does the same thing as `add_handler()` using the `DisconnectHandler`.

`Client.on_raw_update()`

Decorator for handling raw updates.

This does the same thing as `add_handler()` using the `RawUpdateHandler`.

Parameters **group** (`int`, *optional*) – The group identifier, defaults to 0.

4.14 Update Filters

class `pyrogram.Filters`

This class provides access to all library-defined Filters available in Pyrogram.

The Filters listed here are intended to be used with the `MessageHandler` only. At the moment, if you want to filter updates coming from different `Handlers` you have to create your own filters with `Filters.create()` and use them in the same way.

create (*func: callable*, ***kwargs*) → *type*

Create a Filter.

Custom filters give you extra control over which updates are allowed or not to be processed by your handlers.

Parameters

- **name** (`str`) – Your filter’s name. Can be anything you like.
- **func** (*callable*) – A function that accepts two arguments (*filter*, *update*) and returns a Boolean: True if the update should be handled, False otherwise. The “update” argument type will vary depending on which `Handler` is coming from. For example, in a `MessageHandler` the update type will be a `Message`; in a `CallbackQueryHandler` the update type will be a `CallbackQuery`. Your function body can then access the incoming update and decide whether to allow it or not.
- ****kwargs** (*any*, *optional*) – Any keyword argument you would like to pass. Useful for custom filters that accept parameters (e.g.: `Filters.command()`, `Filters.regex()`).

me = `<pyrogram.client.filters.filters.Me object>`

Filter messages generated by you yourself.

bot = `<pyrogram.client.filters.filters.Bot object>`

Filter messages coming from bots.

incoming = <pyrogram.client.filters.filters.Incoming object>
Filter incoming messages. Messages sent to your own chat (Saved Messages) are also recognised as incoming.

outgoing = <pyrogram.client.filters.filters.Outgoing object>
Filter outgoing messages. Messages sent to your own chat (Saved Messages) are not recognized as outgoing.

text = <pyrogram.client.filters.filters.Text object>
Filter text messages.

reply = <pyrogram.client.filters.filters.Reply object>
Filter messages that are replies to other messages.

forwarded = <pyrogram.client.filters.filters.Forwarded object>
Filter messages that are forwarded.

caption = <pyrogram.client.filters.filters.Caption object>
Filter media messages that contain captions.

edited = <pyrogram.client.filters.filters.Edited object>
Filter edited messages.

audio = <pyrogram.client.filters.filters.Audio object>
Filter messages that contain *Audio* objects.

document = <pyrogram.client.filters.filters.Document object>
Filter messages that contain *Document* objects.

photo = <pyrogram.client.filters.filters.Photo object>
Filter messages that contain *Photo* objects.

sticker = <pyrogram.client.filters.filters.Sticker object>
Filter messages that contain *Sticker* objects.

animation = <pyrogram.client.filters.filters.Animation object>
Filter messages that contain *Animation* objects.

game = <pyrogram.client.filters.filters.Game object>
Filter messages that contain *Game* objects.

video = <pyrogram.client.filters.filters.Video object>
Filter messages that contain *Video* objects.

media_group = <pyrogram.client.filters.filters.MediaGroup object>
Filter messages containing photos or videos being part of an album.

voice = <pyrogram.client.filters.filters.Voice object>
Filter messages that contain *Voice* note objects.

video_note = <pyrogram.client.filters.filters.VideoNote object>
Filter messages that contain *VideoNote* objects.

contact = <pyrogram.client.filters.filters.Contact object>
Filter messages that contain *Contact* objects.

location = <pyrogram.client.filters.filters.Location object>
Filter messages that contain *Location* objects.

venue = <pyrogram.client.filters.filters.Venue object>
Filter messages that contain *Venue* objects.

web_page = <pyrogram.client.filters.filters.WebPage object>
Filter messages sent with a webpage preview.

```

poll = <pyrogram.client.filters.filters.Poll object>
    Filter messages that contain Poll objects.

private = <pyrogram.client.filters.filters.Private object>
    Filter messages sent in private chats.

group = <pyrogram.client.filters.filters.Group object>
    Filter messages sent in group or supergroup chats.

channel = <pyrogram.client.filters.filters.Channel object>
    Filter messages sent in channels.

new_chat_members = <pyrogram.client.filters.filters.NewChatMembers object>
    Filter service messages for new chat members.

left_chat_member = <pyrogram.client.filters.filters.LeftChatMember object>
    Filter service messages for members that left the chat.

new_chat_title = <pyrogram.client.filters.filters.NewChatTitle object>
    Filter service messages for new chat titles.

new_chat_photo = <pyrogram.client.filters.filters.NewChatPhoto object>
    Filter service messages for new chat photos.

delete_chat_photo = <pyrogram.client.filters.filters.DeleteChatPhoto object>
    Filter service messages for deleted photos.

group_chat_created = <pyrogram.client.filters.filters.GroupChatCreated object>
    Filter service messages for group chat creations.

supergroup_chat_created = <pyrogram.client.filters.filters.SupergroupChatCreated object>
    Filter service messages for supergroup chat creations.

channel_chat_created = <pyrogram.client.filters.filters.ChannelChatCreated object>
    Filter service messages for channel chat creations.

migrate_to_chat_id = <pyrogram.client.filters.filters.MigrateToChatId object>
    Filter service messages that contain migrate_to_chat_id.

migrate_from_chat_id = <pyrogram.client.filters.filters.MigrateFromChatId object>
    Filter service messages that contain migrate_from_chat_id.

pinned_message = <pyrogram.client.filters.filters.PinnedMessage object>
    Filter service messages for pinned messages.

game_high_score = <pyrogram.client.filters.filters.GameHighScore object>
    Filter service messages for game high scores.

reply_keyboard = <pyrogram.client.filters.filters.ReplyKeyboard object>
    Filter messages containing reply keyboard markups

inline_keyboard = <pyrogram.client.filters.filters.InlineKeyboard object>
    Filter messages containing inline keyboard markups

mentioned = <pyrogram.client.filters.filters.Mentioned object>
    Filter messages containing mentions

via_bot = <pyrogram.client.filters.filters.ViaBot object>
    Filter messages sent via inline bots

service = <pyrogram.client.filters.filters.Service object>
    Filter service messages.

```

A service message contains any of the following fields set: *left_chat_member*, *new_chat_title*, *new_chat_photo*, *delete_chat_photo*, *group_chat_created*, *supergroup_chat_created*, *channel_chat_created*, *migrate_to_chat_id*, *migrate_from_chat_id*, *pinned_message*, *game_score*.

media = <pyrogram.client.filters.filters.Media object>

Filter media messages.

A media message contains any of the following fields set: *audio*, *document*, *photo*, *sticker*, *video*, *animation*, *voice*, *video_note*, *contact*, *location*, *venue*, *poll*.

static command (*commands: str*, *prefix: str = '/'*, *separator: str = ' '*, *case_sensitive: bool = False*)

Filter commands, i.e.: text messages starting with “/” or any other custom prefix.

Parameters

- **commands** (*str | list*) – The command or list of commands as string the filter should look for. Examples: “start”, [“start”, “help”, “settings”]. When a message text containing a command arrives, the command itself and its arguments will be stored in the *command* field of the *Message*.
- **prefix** (*str | list*, *optional*) – A prefix or a list of prefixes as string the filter should look for. Defaults to “/” (slash). Examples: “.”, “!”, [“/”, “!”, “.”]. Can be None or “” (empty string) to allow commands with no prefix at all.
- **separator** (*str*, *optional*) – The command arguments separator. Defaults to “ ” (white space). Examples: /start first second, /start-first-second, /start.first.second.
- **case_sensitive** (*bool*, *optional*) – Pass True if you want your command(s) to be case sensitive. Defaults to False. Examples: when True, command=“Start” would trigger /Start but not /start.

static regex (*pattern*, *flags: int = 0*)

Filter messages that match a given RegEx pattern.

Parameters

- **pattern** (*str*) – The RegEx pattern as string, it will be applied to the text of a message. When a pattern matches, all the *Match Objects* are stored in the *matches* field of the *Message* itself.
- **flags** (*int*, *optional*) – RegEx flags.

class user (*users: int = None*)

Filter messages coming from one or more users.

You can use *set bound methods* to manipulate the users container.

Parameters users (*int | str | list*) – Pass one or more user ids/usernames to filter users.

For you yourself, “me” or “self” can be used as well. Defaults to None (no users).

class chat (*chats: int = None*)

Filter messages coming from one or more chats.

You can use *set bound methods* to manipulate the chats container.

Parameters chats (*int | str | list*) – Pass one or more chat ids/usernames to filter chats.

For your personal cloud (Saved Messages) you can simply use “me” or “self”. Defaults to None (no chats).

4.15 RPC Errors

All Pyrogram API errors live inside the `errors` sub-package: `pyrogram.errors`. The errors ids listed here are shown as *UPPER_SNAKE_CASE*, but the actual exception names to import from Pyrogram follow the usual *PascalCase* convention.

```
from pyrogram.errors import FloodWait

try:
    ...
except FloodWait as e:
    ...
```

4.15.1 303 - SeeOther

id	message
FILE_MIGRATE_X	The file to be accessed is currently stored in DC{x}
PHONE_MIGRATE_X	The phone number a user is trying to use for authorization is associated with DC{x}
NET- WORK_MIGRATE_X	The source IP address is associated with DC{x} (for registration)
USER_MIGRATE_X	The user whose identity is being used to execute queries is associated with DC{x} (for registration)

4.15.2 400 - BadRequest

id	message
FIRSTNAME_INVALID	The first name is invalid
LASTNAME_INVALID	The last name is invalid
PHONE_NUMBER_INVALID	The phone number is invalid
PHONE_CODE_HASH_EMPTY	phone_code_hash is missing
PHONE_CODE_EMPTY	phone_code is missing
PHONE_CODE_EXPIRED	The confirmation code has expired
PHONE_CODE_INVALID	The confirmation code is invalid
API_ID_INVALID	The api_id/api_hash combination is invalid
PHONE_NUMBER_OCCUPIED	The phone number is already in use
PHONE_NUMBER_UNOCCUPIED	The phone number is not yet being used
USERS_TOO_FEW	Not enough users (to create a chat, for example)
USERS_TOO_MUCH	The maximum number of users has been exceeded (to create a chat, for example)
TYPE_CONSTRUCTOR_INVALID	The type constructor is invalid
FILE_PART_INVALID	The file part number is invalid
FILE_PARTS_INVALID	The number of file parts is invalid
FILE_PART_X_MISSING	Part {x} of the file is missing from storage
MD5_CHECKSUM_INVALID	The MD5 checksums do not match
PHOTO_INVALID_DIMENSIONS	The photo dimensions are invalid
FIELD_NAME_INVALID	The field with the name FIELD_NAME is invalid
FIELD_NAME_EMPTY	The field with the name FIELD_NAME is missing
MSG_WAIT_FAILED	A waiting call returned an error
PEER_ID_INVALID	The id/access_hash combination is invalid

Continued on next page

Table 16 – continued from previous page

id	message
MESSAGE_EMPTY	The message sent is empty
ENCRYPTED_MESSAGE_INVALID	The special binding message (bind_auth_key_inner) contains invalid data
INPUT_METHOD_INVALID	The method called is invalid
PASSWORD_HASH_INVALID	Two-step verification password is invalid
USERNAME_NOT_OCCUPIED	The username is not occupied by anyone
USERNAME_INVALID	The username is invalid
MESSAGE_ID_INVALID	The message id is invalid
MESSAGE_NOT_MODIFIED	The message was not modified
ENTITY_MENTION_USER_INVALID	The mentioned entity is not an user
MESSAGE_TOO_LONG	The message text is over 4096 characters
ACCESS_TOKEN_EXPIRED	The bot token is invalid
BOT_METHOD_INVALID	The method can't be used by bots
QUERY_TOO_SHORT	The query is too short
SEARCH_QUERY_EMPTY	The query is empty
CHAT_ID_INVALID	The chat id is invalid
DATE_EMPTY	The date argument is empty
PERSISTENT_TIMESTAMP_EMPTY	The pts is empty
CDN_METHOD_INVALID	The method can't be used on CDN DCs
VOLUME_LOC_NOT_FOUND	The volume location can't be found
FILE_ID_INVALID	The file id is invalid
LOCATION_INVALID	The file location is invalid
CHAT_ADMIN_REQUIRED	The method requires chat admin privileges
PHONE_NUMBER_BANNED	The phone number is banned
ABOUT_TOO_LONG	The about text is too long
MULTI_MEDIA_TOO_LONG	The album contains more than 10 items
USERNAME_OCCUPIED	The username is already in use
BOT_INLINE_DISABLED	The inline feature of the bot is disabled
INLINE_RESULT_EXPIRED	The inline bot query expired
INVITE_HASH_INVALID	The invite link hash is invalid
USER_ALREADY_PARTICIPANT	The user is already a participant of this chat
TTL_MEDIA_INVALID	The media does not support self-destruction
MAX_ID_INVALID	The max_id parameter is invalid
CHANNEL_INVALID	The channel parameter is invalid
DC_ID_INVALID	The dc_id parameter is invalid
LIMIT_INVALID	The limit parameter is invalid
OFFSET_INVALID	The offset parameter is invalid
EMAIL_INVALID	The email provided is invalid
USER_IS_BOT	A bot cannot send messages to other bots or to itself
WEBPAGE_CURL_FAILED	Telegram server could not fetch the provided URL
STICKERSET_INVALID	The requested sticker set is invalid
PEER_FLOOD	The method can't be used because your account is limited
MEDIA_CAPTION_TOO_LONG	The media caption is longer than 1024 characters
USER_NOT_MUTUAL_CONTACT	The user is not a mutual contact
USER_CHANNELS_TOO_MUCH	The user is already in too many channels or supergroups
API_ID_PUBLISHED_FLOOD	You are using an API key that is limited on the server side
USER_NOT_PARTICIPANT	The user is not a member of this chat
CHANNEL_PRIVATE	The channel/supergroup is not accessible
MESSAGE_IDS_EMPTY	The requested message doesn't exist
WEBPAGE_MEDIA_EMPTY	The URL doesn't contain any valid media

Continued on next page

Table 16 – continued from previous page

id	message
QUERY_ID_INVALID	The callback query id is invalid
MEDIA_EMPTY	The media is invalid
USER_IS_BLOCKED	The user blocked you
YOU_BLOCKED_USER	You blocked this user
ADMINS_TOO_MUCH	The chat has too many administrators
BOTS_TOO_MUCH	The chat has too many bots
USER_ADMIN_INVALID	The action requires admin privileges
INPUT_USER_DEACTIVATED	The target user has been deactivated
PASSWORD_RECOVERY_NA	The password recovery e-mail is not available
PASSWORD_EMPTY	The password entered is empty
PHONE_NUMBER_FLOOD	This number has tried to login too many times
TAKEOUT_INVALID	The takeout id is invalid
TAKEOUT_REQUIRED	The method must be invoked inside a takeout session
MESSAGE_POLL_CLOSED	You can't interact with a closed poll
MEDIA_INVALID	The media is invalid
BOT_SCORE_NOT_MODIFIED	The bot score was not modified
USER_BOT_REQUIRED	The method can be used by bots only
IMAGE_PROCESS_FAILED	The server failed to process your image
USERNAME_NOT_MODIFIED	The username was not modified
CALL_ALREADY_ACCEPTED	The call is already accepted
CALL_ALREADY_DECLINED	The call is already declined
PHOTO_EXT_INVALID	The photo extension is invalid
EXTERNAL_URL_INVALID	The external media URL is invalid
CHAT_NOT_MODIFIED	The chat settings were not modified
RESULTS_TOO_MUCH	The result contains too many items
RESULT_ID_DUPLICATE	The result contains items with duplicated identifiers
ACCESS_TOKEN_INVALID	The bot access token is invalid
INVITE_HASH_EXPIRED	The chat invite link is no longer valid
USER_BANNED_IN_CHANNEL	You are limited, check @SpamBot for details

4.15.3 401 - Unauthorized

id	message
AUTH_KEY_UNREGISTERED	The key is not registered in the system
AUTH_KEY_INVALID	The key is invalid
USER_DEACTIVATED	The user has been deleted/deactivated
SESSION_REVOKED	The authorization has been invalidated, because of the user terminating all sessions
SESSION_EXPIRED	The authorization has expired
ACTIVE_USER_REQUIRED	The method is only available to already activated users
AUTH_KEY_PERM_EMPTY	The method is unavailable for temporary authorization key, not bound to permanent
SESSION_PASSWORD_NEEDED	Two-step verification password required

4.15.4 403 - Forbidden

id	message
CHAT_WRITE_FORBIDDEN	You don't have rights to send messages in this chat
RIGHT_FORBIDDEN	One or more admin rights can't be applied to this kind of chat (channel/supergroup)
CHAT_ADMIN_INVITE_REQUIRED	You don't have rights to invite other users
MES-SAGE_DELETE_FORBIDDEN	You don't have rights to delete messages in this chat
CHAT_SEND_MEDIA_FORBIDDEN	You can't send media messages in this chat
MES-SAGE_AUTHOR_REQUIRED	You are not the author of this message

4.15.5 406 - NotAcceptable

id	message
AUTH_KEY_DUPLICATED	Authorization error - you must delete your session file and log in again with your phone number
FIL-EREF_UPGRADE_NEEDED	The file reference has expired - you must obtain the original media message

4.15.6 420 - Flood

id	message
FLOOD_WAIT_X	A wait of {x} seconds is required
TAKE-OUT_INIT_DELAY_X	You have to confirm the data export request using one of your mobile devices or wait {x} seconds

4.15.7 500 - InternalServerError

id	message
AUTH_RESTART	User authorization has restarted
RPC_CALL_FAIL	Telegram is having internal problems. Please try again later
RPC_MCGET_FAIL	Telegram is having internal problems. Please try again later
PERSISTENT_TIMESTAMP_OUTDATED	Telegram is having internal problems. Please try again later
HISTORY_GET_FAILED	Telegram is having internal problems. Please try again later
REG_ID_GENERATE_FAILED	Telegram is having internal problems. Please try again later
RANDOM_ID_DUPLICATE	Telegram is having internal problems. Please try again later
WORKER_BUSY_TOO_LONG_RETRY	Telegram is having internal problems. Please try again later
INTERDC_X_CALL_ERROR	Telegram is having internal problems. Please try again later
INTERDC_X_CALL_RICH_ERROR	Telegram is having internal problems. Please try again later

4.16 Using Filters

So far we've seen how to register a callback function that executes every time a specific update comes from the server, but there's much more than that to come.

Here we'll discuss about *Filters*. Filters enable a fine-grain control over what kind of updates are allowed or not to be passed in your callback functions, based on their inner details.

4.16.1 Single Filters

Let's start right away with a simple example:

- This example will show you how to **only** handle messages containing an *Audio* object and ignore any other message. Filters are passed as the first argument of the decorator:

```
from pyrogram import Filters

@app.on_message(Filters.audio)
def my_handler(client, message):
    print(message)
```

- or, without decorators. Here filters are passed as the second argument of the handler constructor:

```
from pyrogram import Filters, MessageHandler

def my_handler(client, message):
    print(message)

app.add_handler(MessageHandler(my_handler, Filters.audio))
```

4.16.2 Combining Filters

Filters can also be used in a more advanced way by inverting and combining more filters together using bitwise operators `~`, `&` and `|`:

- Use `~` to invert a filter (behaves like the `not` operator).
- Use `&` and `|` to merge two filters (behave like `and`, `or` operators respectively).

Here are some examples:

- Message is a **text** message **and** is **not edited**.

```
@app.on_message(Filters.text & ~Filters.edited)
def my_handler(client, message):
    print(message)
```

- Message is a **sticker** **and** is coming from a **channel** **or** a **private** chat.

```
@app.on_message(Filters.sticker & (Filters.channel | Filters.private))
def my_handler(client, message):
    print(message)
```

4.16.3 Advanced Filters

Some filters, like `command()` or `regex()` can also accept arguments:

- Message is either a `/start` or `/help` **command**.

```
@app.on_message(Filters.command(["start", "help"]))
def my_handler(client, message):
    print(message)
```

- Message is a **text** message or a media **caption** matching the given **regex** pattern.

```
@app.on_message(Filters.regex("pyrogram"))
def my_handler(client, message):
    print(message)
```

More handlers using different filters can also live together.

```
@app.on_message(Filters.command("start"))
def start_command(client, message):
    print("This is the /start command")

@app.on_message(Filters.command("help"))
def help_command(client, message):
    print("This is the /help command")

@app.on_message(Filters.chat("PyrogramChat"))
def from_pyrogramchat(client, message):
    print("New message in @PyrogramChat")
```

4.16.4 Custom Filters

Pyrogram already provides lots of built-in *Filters* to work with, but in case you can't find a specific one for your needs or want to build a custom filter by yourself (to be used in a different kind of handler, for example) you can use `Filters.create()`.

Note: At the moment, the built-in filters are intended to be used with the *MessageHandler* only.

An example to demonstrate how custom filters work is to show how to create and use one for the *CallbackQueryHandler*. Note that callback queries updates are only received by bots; create and `authorize` your bot, then send a message with an inline keyboard to yourself. This allows you to test your filter by pressing the inline button:

```
from pyrogram import InlineKeyboardMarkup, InlineKeyboardButton

app.send_message(
    "username", # Change this to your username or id
    "Pyrogram's custom filter test",
    reply_markup=InlineKeyboardMarkup(
        [[InlineKeyboardButton("Press me", b"pyrogram")]]
    )
)
```

Basic Filters

For this basic filter we will be using only the first two parameters of `Filters.create()`.

The code below creates a simple filter for hardcoded, static callback data. This filter will only allow callback queries containing “Pyrogram” as data, that is, the function *func* you pass returns True in case the callback query data equals to `b"Pyrogram"`.

```
static_data = Filters.create(
    name="StaticData",
    func=lambda flt, callback_query: callback_query.data == b"Pyrogram"
)
```

The lambda operator in python is used to create small anonymous functions and is perfect for this example, the same could be achieved with a normal function, but we don’t really need it as it makes sense only inside the filter’s scope:

```
def func(flt, callback_query):
    return callback_query.data == b"Pyrogram"

static_data = Filters.create(
    name="StaticData",
    func=func
)
```

The filter usage remains the same:

```
@app.on_callback_query(static_data)
def pyrogram_data(client, callback_query):
    client.answer_callback_query(callback_query.id, "it works!")
```

Filters with Arguments

A much cooler filter would be one that accepts “Pyrogram” or any other data as argument at usage time. A dynamic filter like this will make use of the third parameter of `Filters.create()`.

This is how a dynamic custom filter looks like:

```
def dynamic_data(data):
    return Filters.create(
        name="DynamicData",
        func=lambda flt, callback_query: flt.data == callback_query.data,
        data=data # "data" kwarg is accessed with "filter.data"
    )
```

And its usage:

```
@app.on_callback_query(dynamic_data(b"Pyrogram"))
def pyrogram_data(client, callback_query):
    client.answer_callback_query(callback_query.id, "it works!")
```

4.17 More on Updates

Here we’ll show some advanced usages when working with [update handlers](#) and [filters](#).

4.17.1 Handler Groups

If you register handlers with overlapping (conflicting) filters, only the first one is executed and any other handler will be ignored. This is intended by design.

In order to handle the very same update more than once, you have to register your handler in a different dispatching group. Dispatching groups hold one or more handlers and are processed sequentially, they are identified by a number (number 0 being the default) and sorted, that is, a lower group number has a higher priority:

For example, take these two handlers:

```
@app.on_message(Filters.text | Filters.sticker)
def text_or_sticker(client, message):
    print("Text or Sticker")

@app.on_message(Filters.text)
def just_text(client, message):
    print("Just Text")
```

Here, `just_text` is never executed because `text_or_sticker`, which has been registered first, already handles texts (`Filters.text` is shared and conflicting). To enable it, register the function using a different group:

```
@app.on_message(Filters.text, group=1)
def just_text(client, message):
    print("Just Text")
```

Or, if you want `just_text` to be fired *before* `text_or_sticker` (note `-1`, which is less than 0):

```
@app.on_message(Filters.text, group=-1)
def just_text(client, message):
    print("Just Text")
```

With `add_handler()` (without decorators) the same can be achieved with:

```
app.add_handler(MessageHandler(just_text, Filters.text), -1)
```

4.17.2 Update propagation

Registering multiple handlers, each in a different group, becomes useful when you want to handle the same update more than once. Any incoming update will be sequentially processed by all of your registered functions by respecting the groups priority policy described above. Even in case any handler raises an unhandled exception, Pyrogram will still continue to propagate the same update to the next groups until all the handlers are done. Example:

```
@app.on_message(Filters.private)
def _(client, message):
    print(0)

@app.on_message(Filters.private, group=1)
def _(client, message):
    raise Exception("Unhandled exception!") # Simulate an unhandled exception

@app.on_message(Filters.private, group=2)
def _(client, message):
    print(2)
```

All these handlers will handle the same kind of messages, that are, messages sent or received in private chats. The output for each incoming update will therefore be:

```
0
Exception: Unhandled exception!
2
```

Stop Propagation

In order to prevent further propagation of an update in the dispatching phase, you can do *one* of the following:

- Call the update's bound-method `.stop_propagation()` (preferred way).
- Manually raise `StopPropagation` exception (more suitable for raw updates only).

Note: Internally, the propagation is stopped by handling a custom exception. `.stop_propagation()` is just an elegant and intuitive way to raise `StopPropagation`; this also means that any code coming *after* calling the method won't be executed as your function just raised an exception to signal the dispatcher not to propagate the update anymore.

Example with `stop_propagation()`:

```
@app.on_message(Filters.private)
def _(client, message):
    print(0)

@app.on_message(Filters.private, group=1)
def _(client, message):
    print(1)
    message.stop_propagation()

@app.on_message(Filters.private, group=2)
def _(client, message):
    print(2)
```

Example with `raise StopPropagation`:

```
from pyrogram import StopPropagation

@app.on_message(Filters.private)
def _(client, message):
    print(0)

@app.on_message(Filters.private, group=1)
def _(client, message):
    print(1)
    raise StopPropagation

@app.on_message(Filters.private, group=2)
def _(client, message):
    print(2)
```

Each handler is registered in a different group, but the handler in group number 2 will never be executed because the propagation was stopped earlier. The output of both (equivalent) examples will be:

```
0
1
```

Continue Propagation

As opposed to *stopping the update propagation* and also as an alternative to the *handler groups*, you can signal the internal dispatcher to continue the update propagation within **the same group** despite having conflicting filters in the next registered handler. This allows you to register multiple handlers with overlapping filters in the same group; to let the dispatcher process the next handler you can do *one* of the following in each handler you want to grant permission to continue:

- Call the update's bound-method `.continue_propagation()` (preferred way).
- Manually raise `ContinuePropagation` exception (more suitable for raw updates only).

Note: Internally, the propagation is continued by handling a custom exception. `.continue_propagation()` is just an elegant and intuitive way to raise `ContinuePropagation`; this also means that any code coming *after* calling the method won't be executed as your function just raised an exception to signal the dispatcher to continue with the next available handler.

Example with `continue_propagation()`:

```
@app.on_message(Filters.private)
def _(client, message):
    print(0)
    message.continue_propagation()

@app.on_message(Filters.private)
def _(client, message):
    print(1)
    message.continue_propagation()

@app.on_message(Filters.private)
def _(client, message):
    print(2)
```

Example with `raise ContinuePropagation`:

```
from pyrogram import ContinuePropagation

@app.on_message(Filters.private)
def _(client, message):
    print(0)
    raise ContinuePropagation

@app.on_message(Filters.private)
def _(client, message):
    print(1)
    raise ContinuePropagation
```

(continues on next page)

(continued from previous page)

```
@app.on_message(Filters.private)
def _(client, message):
    print(2)
```

Three handlers are registered in the same group, and all of them will be executed because the propagation was continued in each handler (except in the last one, where is useless to do so since there is no more handlers after). The output of both (equivalent) examples will be:

```
0
1
2
```

4.18 Configuration File

As already mentioned in previous sections, Pyrogram can be configured by the use of an INI file. This page explains how this file is structured in Pyrogram, how to use it and why.

4.18.1 Introduction

The idea behind using a configuration file is to help keeping your code free of private settings information such as the API Key and Proxy without having you to even deal with how to load such settings. The configuration file, usually referred as `config.ini` file, is automatically loaded from the root of your working directory; all you need to do is fill in the necessary parts.

Note: The configuration file is optional, but recommended. If, for any reason, you prefer not to use it, there's always an alternative way to configure Pyrogram via Client's parameters. Doing so, you can have full control on how to store and load your settings (e.g.: from environment variables).

Settings specified via Client's parameter have higher priority and will override any setting stored in the configuration file.

4.18.2 The config.ini File

By default, Pyrogram will look for a file named `config.ini` placed at the root of your working directory, that is, the same folder of your running script. You can change the name or location of your configuration file by specifying it in your Client's parameter `config_file`.

- Replace the default `config.ini` file with `my_configuration.ini`:

```
from pyrogram import Client

app = Client("my_account", config_file="my_configuration.ini")
```

4.18.3 Configuration Sections

These are all the sections Pyrogram uses in its configuration file:

Pyrogram

The [pyrogram] section contains your Telegram API credentials: *api_id* and *api_hash*.

```
[pyrogram]
api_id = 12345
api_hash = 0123456789abcdef0123456789abcdef
```

More info about API Key.

Proxy

The [proxy] section contains settings about your SOCKS5 proxy.

```
[proxy]
enabled = True
hostname = 11.22.33.44
port = 1080
username = <your_username>
password = <your_password>
```

More info about SOCKS5 Proxy.

Plugins

The [plugins] section contains settings about Smart Plugins.

```
[plugins]
root = plugins
include =
    module
    folder.module
exclude =
    module fn2
```

More info about Smart Plugins.

4.19 Smart Plugins

Pyrogram embeds a **smart**, lightweight yet powerful plugin system that is meant to further simplify the organization of large projects and to provide a way for creating pluggable (modular) components that can be **easily shared** across different Pyrogram applications with **minimal boilerplate code**.

Tip: Smart Plugins are completely optional and disabled by default.

4.19.1 Introduction

Prior to the Smart Plugin system, pluggable handlers were already possible. For example, if you wanted to modularize your applications, you had to put your function definitions in separate files and register them inside your main script after importing your modules, like this:

Note: This is an example application that replies in private chats with two messages: one containing the same text message you sent and the other containing the reversed text message.

Example: “Pyrogram” replies with “Pyrogram” and “margoryP”

```
myproject/
  config.ini
  handlers.py
  main.py
```

- handlers.py

```
def echo(client, message):
    message.reply(message.text)

def echo_reversed(client, message):
    message.reply(message.text[::-1])
```

- main.py

```
from pyrogram import Client, MessageHandler, Filters

from handlers import echo, echo_reversed

app = Client("my_account")

app.add_handler(
    MessageHandler(
        echo,
        Filters.text & Filters.private))

app.add_handler(
    MessageHandler(
        echo_reversed,
        Filters.text & Filters.private),
    group=1)

app.run()
```

This is already nice and doesn’t add *too much* boilerplate code, but things can get boring still; you have to manually import, manually `add_handler()` and manually instantiate each `MessageHandler` object because **you can’t use those cool decorators** for your functions. So, what if you could? Smart Plugins solve this issue by taking care of handlers registration automatically.

4.19.2 Using Smart Plugins

Setting up your Pyrogram project to accommodate Smart Plugins is pretty straightforward:

1. Create a new folder to store all the plugins (e.g.: “plugins”, “handlers”, ...).
2. Put your python files full of plugins inside. Organize them as you wish.
3. Enable plugins in your Client or via the `config.ini` file.

Note: This is the same example application *as shown above*, written using the Smart Plugin system.

```
myproject/  
  plugins/  
    handlers.py  
  config.ini  
  main.py
```

- plugins/handlers.py

```
from pyrogram import Client, Filters  
  
@Client.on_message(Filters.text & Filters.private)  
def echo(client, message):  
    message.reply(message.text)  
  
@Client.on_message(Filters.text & Filters.private, group=1)  
def echo_reversed(client, message):  
    message.reply(message.text[::-1])
```

- config.ini

```
[plugins]  
root = plugins
```

- main.py

```
from pyrogram import Client  
  
Client("my_account").run()
```

Alternatively, without using the *config.ini* file:

```
from pyrogram import Client  
  
plugins = dict(root="plugins")  
  
Client("my_account", plugins=plugins).run()
```

The first important thing to note is the new `plugins` folder. You can put *any python file* in *any subfolder* and each file can contain *any decorated function* (handlers) with one limitation: within a single module (file) you must use different names for each decorated function.

The second thing is telling Pyrogram where to look for your plugins: you can either use the *config.ini* file or the `Client` parameter “plugins”; the *root* value must match the name of your plugins root folder. Your Pyrogram `Client` instance will **automatically** scan the folder upon starting to search for valid handlers and register them for you.

Then you’ll notice you can now use decorators. That’s right, you can apply the usual decorators to your callback functions in a static way, i.e. **without having the Client instance around**: simply use `@Client` (`Client` class) instead of the usual `@app` (`Client` instance) and things will work just the same.

4.19.3 Specifying the Plugins to include

By default, if you don't explicitly supply a list of plugins, every valid one found inside your plugins root folder will be included by following the alphabetical order of the directory structure (files and subfolders); the single handlers found inside each module will be, instead, loaded in the order they are defined, from top to bottom.

Note: Remember: there can be at most one handler, within a group, dealing with a specific update. Plugins with overlapping filters included a second time will not work. Learn more at [More on Updates](#).

This default loading behaviour is usually enough, but sometimes you want to have more control on what to include (or exclude) and in which exact order to load plugins. The way to do this is to make use of `include` and `exclude` directives, either in the `config.ini` file or in the dictionary passed as `Client` argument. Here's how they work:

- If both `include` and `exclude` are omitted, all plugins are loaded as described above.
- If `include` is given, only the specified plugins will be loaded, in the order they are passed.
- If `exclude` is given, the plugins specified here will be unloaded.

The `include` and `exclude` value is a **list of strings**. Each string containing the path of the module relative to the plugins root folder, in Python notation (dots instead of slashes).

E.g.: `subfolder.module` refers to `plugins/subfolder/module.py`, with `root="plugins"`.

You can also choose the order in which the single handlers inside a module are loaded, thus overriding the default top-to-bottom loading policy. You can do this by appending the name of the functions to the module path, each one separated by a blank space.

E.g.: `subfolder.module fn2 fn1 fn3` will load `fn2`, `fn1` and `fn3` from `subfolder.module`, in this order.

Examples

Given this plugins folder structure with three modules, each containing their own handlers (`fn1`, `fn2`, etc...), which are also organized in subfolders:

```
myproject/
  plugins/
    subfolder1/
      plugins1.py
        - fn1
        - fn2
        - fn3
    subfolder2/
      plugins2.py
    ...
  plugins0.py
  ...
```

- Load every handler from every module, namely `plugins0.py`, `plugins1.py` and `plugins2.py` in alphabetical order (files) and definition order (handlers inside files):

Using `config.ini` file:

```
[plugins]
root = plugins
```

Using *Client*'s parameter:

```
plugins = dict(root="plugins")

Client("my_account", plugins=plugins).run()
```

- Load only handlers defined inside *plugins2.py* and *plugins0.py*, in this order:

Using *config.ini* file:

```
[plugins]
root = plugins
include =
    subfolder2.plugins2
    plugins0
```

Using *Client*'s parameter:

```
plugins = dict(
    root="plugins",
    include=[
        "subfolder2.plugins2",
        "plugins0"
    ]
)

Client("my_account", plugins=plugins).run()
```

- Load everything except the handlers inside *plugins2.py*:

Using *config.ini* file:

```
[plugins]
root = plugins
exclude = subfolder2.plugins2
```

Using *Client*'s parameter:

```
plugins = dict(
    root="plugins",
    exclude=["subfolder2.plugins2"]
)

Client("my_account", plugins=plugins).run()
```

- Load only *fn3*, *fn1* and *fn2* (in this order) from *plugins1.py*:

Using *config.ini* file:

```
[plugins]
root = plugins
include = subfolder1.plugins1 fn3 fn1 fn2
```

Using *Client*'s parameter:

```
plugins = dict(
    root="plugins",
    include=["subfolder1.plugins1 fn3 fn1 fn2"]
)

Client("my_account", plugins=plugins).run()
```

4.19.4 Load/Unload Plugins at Runtime

In the [previous section](#) we've explained how to specify which plugins to load and which to ignore before your Client starts. Here we'll show, instead, how to unload and load again a previously registered plugin at runtime.

Each function decorated with the usual `on_message` decorator (or any other decorator that deals with Telegram updates) will be modified in such a way that, when you reference them later on, they will be actually pointing to a tuple of (*handler: Handler, group: int*). The actual callback function is therefore stored inside the handler's *callback* attribute. Here's an example:

- `plugins/handlers.py`

```
@Client.on_message(Filters.text & Filters.private)
def echo(client, message):
    message.reply(message.text)

print(echo)
print(echo[0].callback)
```

- Printing `echo` will show something like `<MessageHandler object at 0x10e3abc50>, 0`.
- Printing `echo[0].callback`, that is, the *callback* attribute of the first element of the tuple, which is an *Handler*, will reveal the actual callback `<function echo at 0x10e3b6598>`.

Unloading

In order to unload a plugin, or any other handler, all you need to do is obtain a reference to it by importing the relevant module and call `remove_handler()` Client's method with your function name preceded by the star `*` operator as argument. Example:

- `main.py`

```
from plugins.handlers import echo

...

app.remove_handler(*echo)
```

The star `*` operator is used to unpack the tuple into positional arguments so that `remove_handler` will receive exactly what is needed. The same could have been achieved with:

```
handler, group = echo
app.remove_handler(handler, group)
```

Loading

Similarly to the unloading process, in order to load again a previously unloaded plugin you do the same, but this time using `add_handler()` instead. Example:

- main.py

```
from plugins.handlers import echo

...

app.add_handler(*echo)
```

4.20 Auto Authorization

Manually writing phone number, phone code and password on the terminal every time you want to login can be tedious. Pyrogram is able to automate both **Log In** and **Sign Up** processes, all you need to do is pass the relevant parameters when creating a new *Client*.

Note: If you omit any of the optional parameter required for the authorization, Pyrogram will ask you to manually write it. For instance, if you don't want to set a `last_name` when creating a new account you have to explicitly pass an empty string `""`; the default value (`None`) will trigger the `input()` call.

4.20.1 Log In

To automate the **Log In** process, pass your `phone_number` and `password` (if you have one) in the *Client* parameters. If you want to retrieve the phone code programmatically, pass a callback function in the `phone_code` field — this function accepts a single positional argument (`phone_number`) and must return the correct phone code (e.g., “12345”) — otherwise, ignore this parameter, Pyrogram will ask you to input the phone code manually.

Example:

```
from pyrogram import Client

def phone_code_callback(phone_number):
    code = ... # Get your code programmatically
    return code # e.g., "12345"

app = Client(
    session_name="example",
    phone_number="39*****",
    phone_code=phone_code_callback, # Note the missing parentheses
    password="password" # (if you have one)
)

with app:
    print(app.get_me())
```

4.20.2 Sign Up

To automate the **Sign Up** process (i.e., automatically create a new Telegram account), simply fill **both** `first_name` and `last_name` fields alongside the other parameters; they will be used to automatically create a new Telegram account in case the phone number you passed is not registered yet.

Example:


```

from pyrogram import Client

def phone_code_callback(phone_number):
    code = ... # Get your code programmatically
    return code # e.g., "12345"

app = Client(
    session_name="example",
    phone_number="39*****",
    phone_code_callback=phone_code_callback, # Note the missing parentheses
    first_name="Pyrogram",
    last_name="" # Can be an empty string
)

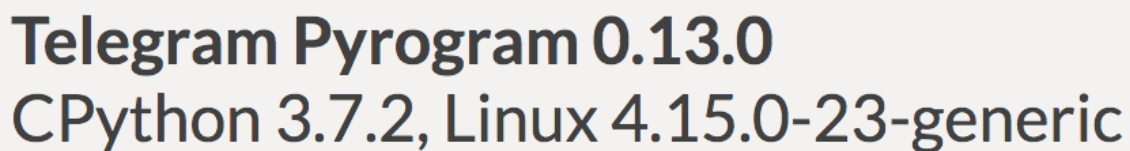
with app:
    print(app.get_me())

```

4.21 Session Settings

As you may probably know, Telegram allows users (and bots) having more than one session (authorizations) registered in the system at the same time.

Briefly explaining, sessions are simply new logins in your account. They can be reviewed in the settings of an official app (or by invoking `GetAuthorizations` with Pyrogram). They store some useful information such as the client who's using them and from which country and IP address.



Telegram Pyrogram 0.13.0
CPython 3.7.2, Linux 4.15.0-23-generic

Fig. 1: A Pyrogram session running on Linux, Python 3.7.

That's how a session looks like on the Android app, showing the three main pieces of information.

- app_version: **Pyrogram 0.13.0**
- device_model: **CPython 3.7.2**
- system_version: **Linux 4.15.0-23-generic**

4.21.1 Set Custom Values

To set custom values, you can either make use of the `config.ini` file, this way:

```

[pyrogram]
app_version = 1.2.3
device_model = PC
system_version = Linux

```

Or, pass the arguments directly in the `Client`'s constructor.

```
app = Client(
    "my_account",
    app_version="1.2.3",
    device_model="PC",
    system_version="Linux"
)
```

4.21.2 Set Custom Languages

To tell Telegram in which language should speak to you (terms of service, bots, service messages, ...) you can set `lang_code` in [ISO 639-1](#) standard (defaults to “en”, English).

With the following code we make Telegram know we want it to speak in Italian (it):

```
[pyrogram]
lang_code = it
```

```
app = Client(
    "my_account",
    lang_code="it",
)
```

4.22 Fast Crypto

Pyrogram’s speed can be *dramatically* boosted up by [TgCrypto](#), a high-performance, easy-to-install Telegram Crypto Library specifically written in C for Pyrogram¹ as a Python extension.

TgCrypto is a replacement for the much slower PyAES and implements the crypto algorithms Telegram requires, namely **AES-IGE 256 bit** (used in MTProto v2.0) and **AES-CTR 256 bit** (used for CDN encrypted files).

4.22.1 Installation

```
$ pip3 install --upgrade tgcrypto
```

Note: Being a C extension for Python, TgCrypto is an optional but *highly recommended* dependency; when TgCrypto is not detected in your system, Pyrogram will automatically fall back to PyAES and will show you a warning.

The reason about being an optional package is that TgCrypto requires some extra system tools in order to be compiled. The errors you receive when trying to install TgCrypto are system dependent, but also descriptive enough to understand what you should do next:

- **Windows:** Install [Visual C++ 2015 Build Tools](#).
- **macOS:** A pop-up will automatically ask you to install the command line developer tools.
- **Linux:** Install a proper C compiler (`gcc`, `clang`) and the Python header files (`python3-dev`).
- **Termux (Android):** Install `clang` and `python-dev` packages.

¹ Although TgCrypto is intended for Pyrogram, it is shipped as a standalone package and can thus be used for other Python projects too.

4.23 Text Formatting

Pyrogram, just like the [Telegram Bot API](#), natively supports basic Markdown and HTML formatting styles for text messages and media captions.

Markdown style uses the same syntax as Telegram Desktop's and is enabled by default.

Beside bold, italic, and pre-formatted code, **Pyrogram does also support inline URLs and inline mentions of users.**

4.23.1 Markdown Style

To use this mode, pass “markdown” in the *parse_mode* field when using `send_message()`. Use the following syntax in your message:

```
**bold text**

__italic text__

[inline URL] (https://docs.pyrogram.org/)

[inline mention of a user] (tg://user?id=23122162)

`inline fixed-width code`

```block_language
pre-formatted fixed-width code block
```
```

4.23.2 HTML Style

To use this mode, pass “html” in the *parse_mode* field when using `send_message()`. The following tags are currently supported:

```
<b>bold</b>, <strong>bold</strong>

<i>italic</i>, <em>italic</em>

<a href="http://docs.pyrogram.org/">inline URL</a>

<a href="tg://user?id=23122162">inline mention of a user</a>

<code>inline fixed-width code</code>

<pre>pre-formatted fixed-width code block</pre>
```

Note: Mentions are only guaranteed to work if you have already met the user (in groups or private chats).

4.23.3 Examples

- Markdown:

```
app.send_message(
    chat_id="haskell",
    text=(
        "**bold**", "
        __italic__, "
        [mention] (tg://user?id=23122162), "
        [URL] (https://docs.pyrogram.org), "
        `code`, "
        ```"
 "for i in range(10):\n"
 " print(i)```"
)
)
```

- HTML:

```
app.send_message(
 chat_id="haskell",
 text=(
 "bold", "
 "<i>italic</i>", "
 "mention", "
 "URL", "
 "<code>code</code>", "
 "<pre>"
 "for i in range(10):\n"
 " print(i)"
 "</pre>"
),
 parse_mode="html"
)
```

## 4.24 Object Serialization

Serializing means converting a Pyrogram object, which exists as Python class instance, to a text string that can be easily shared and stored anywhere. Pyrogram provides two formats for serializing its objects: one good looking for humans and another more compact for machines that is able to recover the original structures.

### 4.24.1 For Humans - `str(obj)`

If you want a nicely formatted, human readable JSON representation of any object in the API – namely, any object from [Pyrogram types](#), [raw functions](#) and [raw types](#) – you can use `str(obj)`.

```
...

with app:
 r = app.get_chat("haskell")

 print(str(r))
```

---

**Tip:** When using `print()` you don't actually need to use `str()` on the object because it is called automatically, we have done that above just to show you how to explicitly convert a Pyrogram object to JSON.

---

### 4.24.2 For Machines - repr(obj)

If you want to share or store objects for future references in a more compact way, you can use `repr(obj)`. While still pretty much readable, this format is not intended for humans. The advantage of this format is that once you serialize your object, you can use `eval()` to get back the original structure; just make sure to `import pyrogram`, as the process requires the package to be in scope.

```
import pyrogram

...

with app:
 r = app.get_chat("haskell")

 print(repr(r))
 print(eval(repr(r)) == r) # True
```

**Note:** Type definitions are subject to changes between versions. You should make sure to store and load objects using the same Pyrogram version.

## 4.25 SOCKS5 Proxy

Pyrogram supports proxies with and without authentication. This feature allows Pyrogram to exchange data with Telegram through an intermediate SOCKS5 proxy server.

### 4.25.1 Usage

- To use Pyrogram with a proxy, simply append the following to your `config.ini` file and replace the values with your own settings:

```
[proxy]
enabled = True
hostname = 11.22.33.44
port = 1080
username = <your_username>
password = <your_password>
```

To enable or disable the proxy without deleting your settings from the config file, change the `enabled` value as follows:

- 1, yes, True or on: Enables the proxy
- 0, no, False or off: Disables the proxy
- Alternatively, you can setup your proxy without the need of the `config.ini` file by using the `proxy` parameter in the `Client` class:

```
from pyrogram import Client

app = Client(
 session_name="example",
 proxy=dict(
```

(continues on next page)

(continued from previous page)

```
 hostname="11.22.33.44",
 port=1080,
 username="<your_username>",
 password="<your_password>"
)
)

app.start()

...
```

---

**Note:** If your proxy doesn't require authorization you can omit `username` and `password` by either leaving the values blank/empty or completely delete the lines.

---

## 4.26 Bots Interaction

Users can interact with other bots via plain text messages as well as inline queries.

### 4.26.1 Inline Bots

- If a bot accepts inline queries, you can call it by using `get_inline_bot_results()` to get the list of its inline results for a query:

```
Get bot results for "Fuzz Universe" from the inline bot @vid
bot_results = app.get_inline_bot_results("vid", "Fuzz Universe")
```

- After you retrieved the bot results, you can use `send_inline_bot_result()` to send a chosen result to any chat:

```
Send the first result to your own chat
app.send_inline_bot_result(
 "me",
 bot_results.query_id,
 bot_results.results[0].id
)
```

## 4.27 MTProto vs. Bot API

Pyrogram is a framework that acts as a fully-fledged Telegram client based on MTProto, and this very feature makes it already superior to, what is usually called, the official Bot API, in many respects. This page will therefore show you why Pyrogram might be a better choice for your project by comparing the two APIs, but first, let's make it clear what actually is the MTProto and the Bot API.

### 4.27.1 What is the MTProto API?

**MTProto**, took alone, is the name of the custom-made, open and encrypted communication protocol created by Telegram itself — it's the only protocol used to exchange information between a client and the actual Telegram servers.

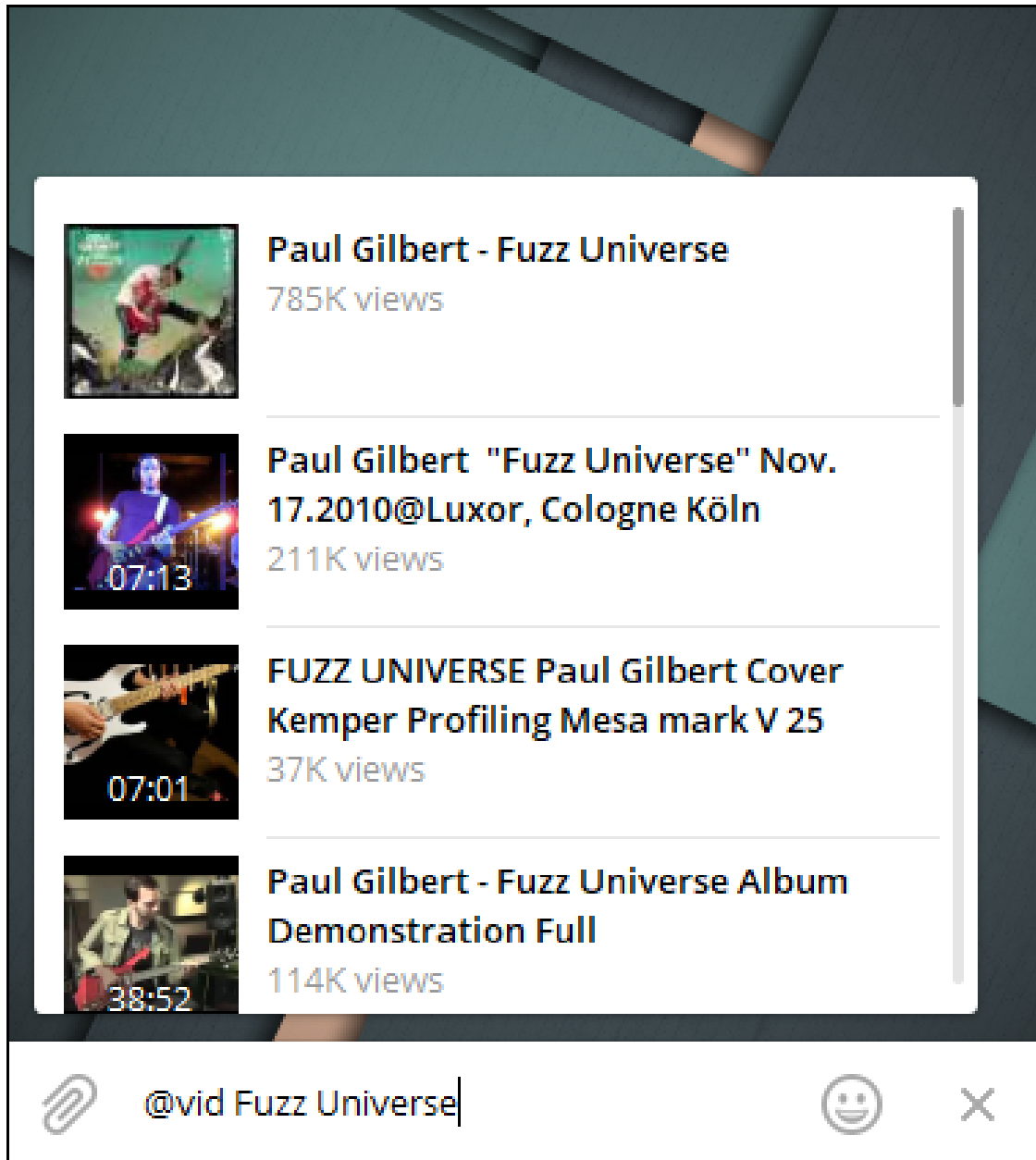


Fig. 2: `get_inline_bot_results()` is the equivalent action of writing `@vid Fuzz Universe` and getting the results list.

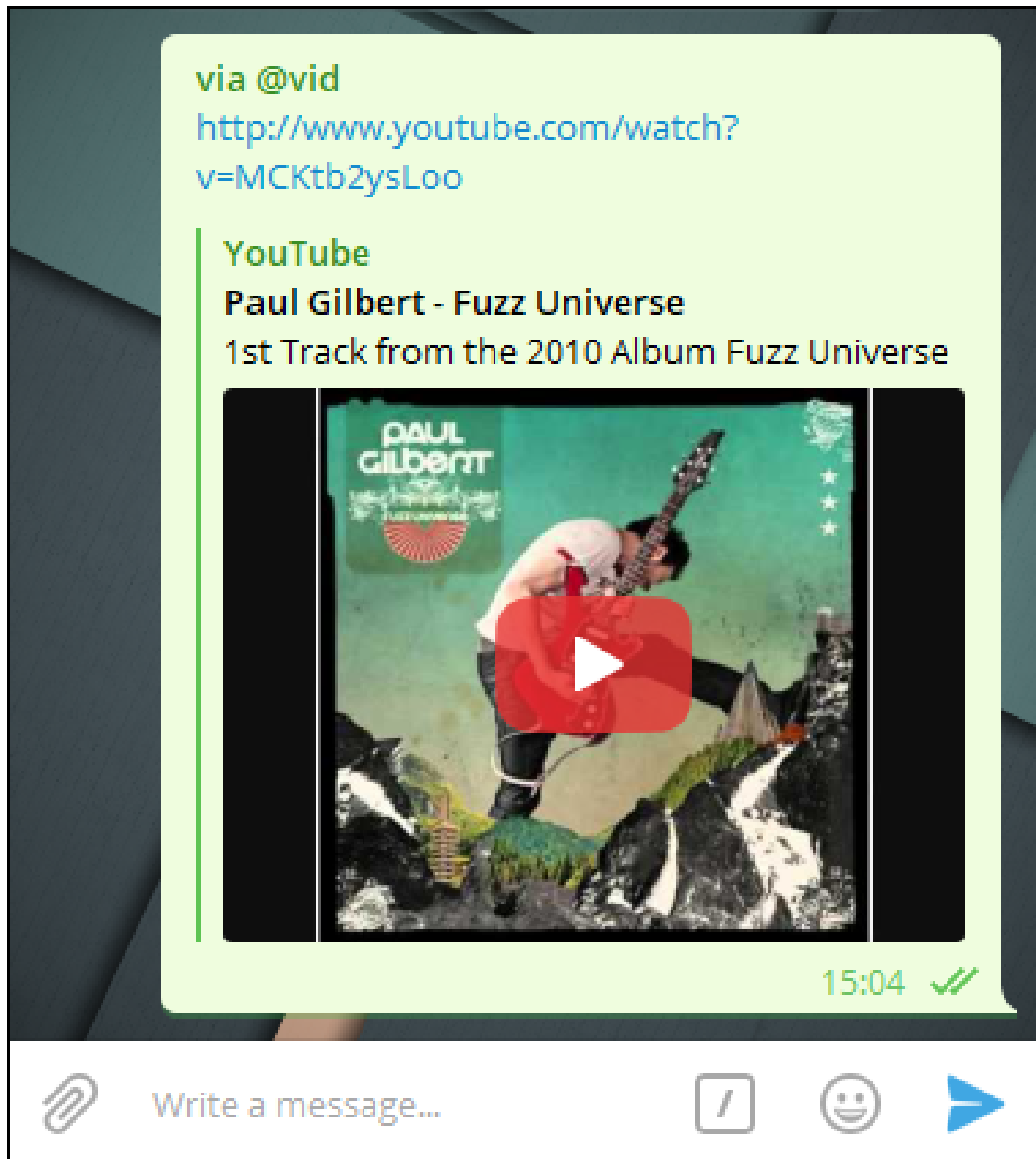


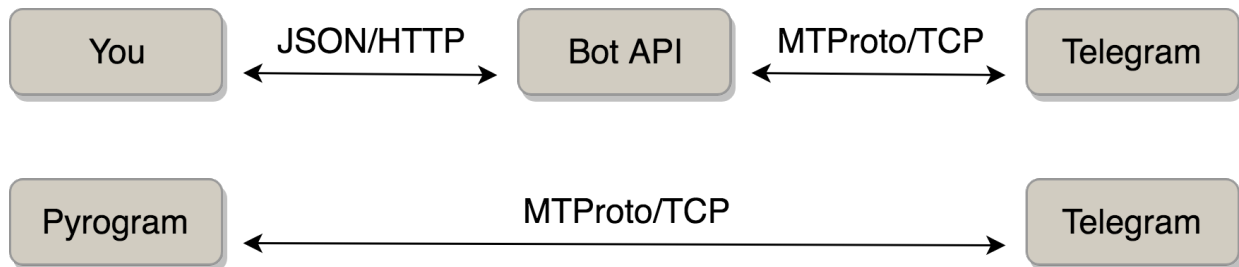
Fig. 3: `send_inline_bot_result()` is the equivalent action of choosing a result from the list and sending it to a chat.



The MTPROTO API on the other hand, is what people, for convenience, call the main Telegram API as a whole. This API is able to authorize both users and bots and is built on top of the MTPROTO encryption protocol by means of [binary data serialized](#) in a specific way, as described by the [TL language](#), and delivered using UDP, TCP or even HTTP as transport-layer protocol.

### 4.27.2 What is the Bot API?

The [Bot API](#) is an HTTP(S) interface for building normal bots using a sub-set of the main MTPROTO API. Bots are special accounts that are authorized via tokens instead of phone numbers. The Bot API is built yet again on top of the main Telegram API, but runs on an intermediate server application that in turn communicates with the actual Telegram servers using MTPROTO.



### 4.27.3 Advantages of the MTPROTO API

Here is a list of all the advantages in using MTPROTO-based libraries – such as Pyrogram – instead of the official HTTP Bot API. Using Pyrogram you can:

- **+ Authorize both user and bot identities**
- – The Bot API only allows bot accounts
- **+ Upload & download any file, up to 1500 MB each (~1.5 GB)**
- – The Bot API allows uploads and downloads of files only up to 50 MB / 20 MB in size (respectively).
- **+ Has less overhead due to direct connections to Telegram**
- – The Bot API uses an intermediate server to handle HTTP requests before they are sent to the actual Telegram servers.
- **+ Run multiple sessions at once, up to 10 per account (either bot or user)**
- – The Bot API intermediate server will terminate any other session in case you try to use the same bot again in a parallel connection.
- **+ Has much more detailed types and powerful methods**
- – The Bot API types often miss some useful information about Telegram entities and some of the methods are limited as well.
- **+ Get information about any public chat by usernames, even if not a member**
- – The Bot API simply doesn't support this
- **+ Obtain information about any message existing in a chat using their ids**
- – The Bot API simply doesn't support this
- **+ Retrieve the whole chat members list of either public or private chats**
- – The Bot API simply doesn't support this
- **+ Receive extra updates, such as the one about a user name change**
- – The Bot API simply doesn't support this

- + **Has more meaningful errors in case something went wrong**
- – The Bot API reports less detailed errors
- + **Get API version updates, and thus new features, sooner**
- – The Bot API is simply slower in implementing new features

## 4.28 Test Servers

If you wish to test your application in a separate environment, Pyrogram is able to authorize your account into Telegram’s test servers without hassle. All you need to do is start a new session (e.g.: “my\_account\_test”) using `test_mode=True`:

```
from pyrogram import Client

with Client("my_account_test", test_mode=True) as app:
 print(app.get_me())
```

---

**Note:** If this is the first time you login into test servers, you will be asked to register your account first. Don’t worry about your contacts and chats, they will be kept untouched inside the production environment; accounts authorized on test servers reside in a different, parallel instance of a Telegram database.

---

### 4.28.1 Test Mode in Official Apps

You can also login yourself into test servers using official desktop apps, such as Webogram and TDesktop:

- **Webogram:** Login here: <https://web.telegram.org/?test=1>
- **TDesktop:** Open settings and type `testmode`.

### 4.28.2 Test Numbers

Beside normal numbers, the test environment allows you to login with reserved test numbers. Valid phone numbers follow the pattern `99966XXXXX`, where `X` is the DC number (1 to 3) and `YYYY` are random numbers. Users with such numbers always get `XXXXXX` as the confirmation code (the DC number, repeated five times).

---

**Important:** Do not store any important or private information in such test users’ accounts; anyone can make use of the simplified authorization mechanism and login at any time.

---

## 4.29 Advanced Usage

Pyrogram’s API, which consists of well documented convenience `methods` and facade `types`, exists to provide a much easier interface to the undocumented and often confusing Telegram API.

In this section, you’ll be shown the alternative way of communicating with Telegram using Pyrogram: the main “raw” Telegram API with its functions and types.

### 4.29.1 Telegram Raw API

If you can't find a high-level method for your needs or if you want complete, low-level access to the whole Telegram API, you have to use the raw *functions* and *types*.

As already hinted, raw functions and types can be really confusing, mainly because people don't realize soon enough they accept *only* the right types and that all required parameters must be filled in. This section will therefore explain some pitfalls to take into consideration when working with the raw API.

---

**Hint:** Every available high-level methods in Pyrogram is built on top of these raw functions.

Nothing stops you from using the raw functions only, but they are rather complex and *plenty of them* are already re-implemented by providing a much simpler and cleaner interface which is very similar to the Bot API (yet much more powerful).

If you think a raw function should be wrapped and added as a high-level method, feel free to ask in our [Community!](#)

---

#### Invoking Functions

Unlike the *methods* found in Pyrogram's API, which can be called in the usual simple way, functions to be invoked from the raw Telegram API have a different way of usage and are more complex.

First of all, both *raw functions* and *raw types* live in their respective packages (and sub-packages): `pyrogram.api.functions`, `pyrogram.api.types`. They all exist as Python classes, meaning you need to create an instance of each every time you need them and fill them in with the correct values using named arguments.

Next, to actually invoke the raw function you have to use the `send()` method provided by the Client class and pass the function object you created.

Here's some examples:

- Update first name, last name and bio:

```
from pyrogram import Client
from pyrogram.api import functions

with Client("my_account") as app:
 app.send(
 functions.account.UpdateProfile(
 first_name="Dan", last_name="Tès",
 about="Bio written from Pyrogram"
)
)
```

- Disable links to your account when someone forwards your messages:

```
from pyrogram import Client
from pyrogram.api import functions, types

with Client("my_account") as app:
 app.send(
 functions.account.SetPrivacy(
 key=types.PrivacyKeyForwards(),
 rules=[types.InputPrivacyValueDisallowAll()]
)
)
```

- Invite users to your channel/supergroup:

```
from pyrogram import Client
from pyrogram.api import functions, types

with Client("my_account") as app:
 app.send(
 functions.channels.InviteToChannel(
 channel=app.resolve_peer(123456789), # ID or Username
 users=[# The users you want to invite
 app.resolve_peer(23456789), # By ID
 app.resolve_peer("username"), # By username
 app.resolve_peer("+393281234567"), # By phone number
]
)
)
```

## Chat IDs

The way Telegram works makes it impossible to directly send a message to a user or a chat by using their IDs only. Instead, a pair of `id` and `access_hash` wrapped in a so called `InputPeer` is always needed. Pyrogram allows sending messages with IDs only thanks to cached access hashes.

There are three different `InputPeer` types, one for each kind of Telegram entity. Whenever an `InputPeer` is needed you must pass one of these:

- `InputPeerUser` - Users
- `InputPeerChat` - Basic Chats
- `InputPeerChannel` - Either Channels or Supergroups

But you don't necessarily have to manually instantiate each object because, luckily for you, Pyrogram already provides `resolve_peer()` as a convenience utility method that returns the correct `InputPeer` by accepting a peer ID only.

Another thing to take into consideration about chat IDs is the way they are represented: they are all integers and all positive within their respective raw types.

Things are different when working with Pyrogram's API because having them in the same space can theoretically lead to collisions, and that's why Pyrogram (as well as the official Bot API) uses a slightly different representation for each kind of ID.

For example, given the ID `123456789`, here's how Pyrogram can tell entities apart:

- `+ID User`: `123456789`
- `-ID Chat`: `-123456789`
- `-100ID Channel (and Supergroup)`: `-100123456789`

So, every time you take a raw ID, make sure to translate it into the correct ID when you want to use it with an high-level method.

## 4.30 Voice Calls

A working proof-of-concept of Telegram voice calls using Pyrogram can be found here: <https://github.com/bakatrouble/pylibtgvoip>. Thanks to @bakatrouble.

---

**Note:** This page will be updated with more information once voice calls become eventually more usable and more integrated in Pyrogram itself.

---

## 4.31 Pyrogram FAQ

This FAQ page provides answers to common questions about Pyrogram and, to some extent, Telegram in general.

---

**Tip:** If you think something interesting could be added here, feel free to propose it by opening a [Feature Request](#).

---

### Contents

- *What is Pyrogram?*
- *Where does the name come from?*
- *How old is Pyrogram?*
- *Why Pyrogram?*
- *What can MTProto do more than the Bot API?*
- *Why do I need an API key for bots?*
- *Can I use the same file\_id across different accounts?*
- *Can I use Bot API's file\_ids in Pyrogram?*
- *Can I use multiple clients at once on the same account?*
- *I started a client and nothing happens!*
- *I keep getting PEER\_ID\_INVALID error!*
- *My account has been deactivated/limited!*
- *About the License*

### 4.31.1 What is Pyrogram?

**Pyrogram** is an elegant, easy-to-use [Telegram](#) client library and framework written from the ground up in Python and C. It enables you to easily create custom applications for both user and bot identities (bot API alternative) via the [MTProto API](#) with the Python programming language.

### 4.31.2 Where does the name come from?

The name “Pyrogram” is composed by **pyro**, which comes from the Greek word  $\pi\rho$  (*pyr*), meaning fire, and **gram**, from *Telegram*. The word *pyro* itself is built from *Python*, **py** for short, and the suffix **ro** to come up with the word *fire*, which also inspired the project logo.

### 4.31.3 How old is Pyrogram?

Pyrogram was first released on December 12, 2017. The actual work on the framework began roughly three months prior the initial public release on [GitHub](#).

### 4.31.4 Why Pyrogram?

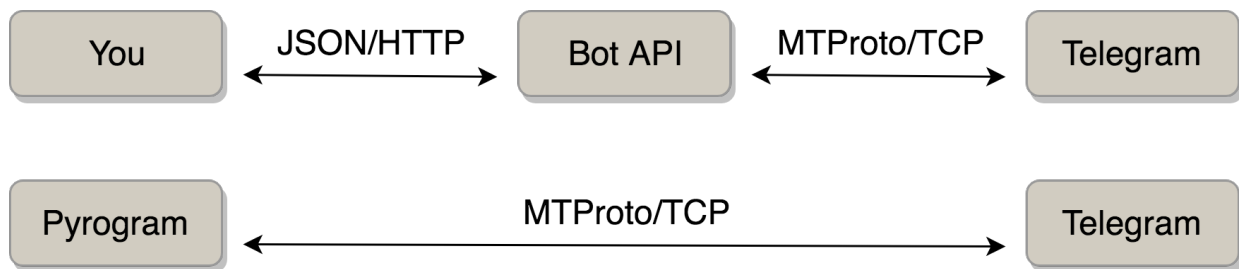
- **Easy:** You can install Pyrogram with pip and start building your applications right away.
- **Elegant:** Low-level details are abstracted and re-presented in a much nicer and easier way.
- **Fast:** Crypto parts are boosted up by [TgCrypto](#), a high-performance library written in pure C.
- **Documented:** Pyrogram API methods, types and public interfaces are well documented.
- **Type-hinted:** Exposed Pyrogram types and method parameters are all type-hinted.
- **Updated,** to make use of the latest Telegram API version and features.
- **Bot API-like:** Similar to the Bot API in its simplicity, but much more powerful and detailed.
- **Pluggable:** The [Smart Plugin](#) system allows to write components with minimal boilerplate code.
- **Comprehensive:** Execute any [advanced action](#) an official client is able to do, and even more.

### 4.31.5 What can MTProto do more than the Bot API?

For a detailed answer, please refer to the [MTProto vs. Bot API](#) page.

### 4.31.6 Why do I need an API key for bots?

Requests against the official bot API endpoint are made via JSON/HTTP, but are handled by an intermediate server application that implements the MTProto protocol – just like Pyrogram – and uses its own API key, which is always required, but hidden to the public.



Using MTProto is the only way to communicate with the actual Telegram servers, and the main API requires developers to identify applications by means of a unique key; the bot token identifies a bot as a user and replaces the user's phone number only.

### 4.31.7 Can I use the same file\_id across different accounts?

No, Telegram doesn't allow this.

File ids are personal and bound to a specific user/bot – and an attempt in using a foreign file id will result in errors such as `[400 MEDIA_EMPTY]`.

The only exception are stickers' file ids; you can use them across different accounts without any problem, like this one: CAADBAADyg4AAvLQYAEYD4F7vcZ43AI.

### 4.31.8 Can I use Bot API's file\_ids in Pyrogram?

Definitely! All file ids you might have taken from the Bot API are 100% compatible and re-usable in Pyrogram. . . at least for now.

Telegram is slowly changing some server's internals and it's doing it in such a way that file ids are going to break inevitably. Not only this, but it seems that the new, hypothetical, file ids could also possibly expire at anytime, thus losing the *persistence* feature.

This change will most likely affect the official [Bot API](#) too (unless Telegram implements some workarounds server-side to keep backwards compatibility, which Pyrogram could in turn make use of) and we can expect a proper notice from Telegram.

### 4.31.9 Can I use multiple clients at once on the same account?

Yes, you can. Both user and bot accounts are able to run multiple sessions in parallel (up to 10 per account). However, you must pay attention and not use the *same* exact session in more than one client at the same time. In other words:

- Avoid copying your session file: even if you rename the file, the copied sessions will still point to a specific one stored in the server.
- Make sure that only one instance of your script runs, using your session file.

If you – even accidentally – fail to do so, all the previous session copies will immediately stop receiving updates and eventually the server will start throwing the error `[406 AUTH_KEY_DUPLICATED]`, inviting you to login again.

Why is that so? Because the server has recognized two identical sessions are running in two different locations, and concludes it could possibly be due to a cloned/stolen device. Having the session ended in such occasions will protect the user's privacy.

So, the only correct way to run multiple clients on the same account is authorizing your account (either user or bot) from the beginning every time, and use one separate session for each parallel client you are going to use.

### 4.31.10 I started a client and nothing happens!

If you are connecting from Russia, China or Iran [you need a proxy](#), because Telegram could be partially or totally blocked in those countries.

Another possible cause might be network issues, either yours or Telegram's. To confirm this, add the following code on the top of your script and run it again. You should see some error mentioning a socket timeout or an unreachable network in a bunch of seconds:

```
import logging
logging.basicConfig(level=logging.INFO)
```

Another way to confirm you aren't able to connect to Telegram is by pinging these IP addresses and see whether ping fails or not:

- DC1: 149.154.175.50
- DC2: 149.154.167.51
- DC3: 149.154.175.100

- DC4: 149.154.167.91
- DC5: 91.108.56.149

#### 4.31.11 I keep getting PEER\_ID\_INVALID error!

The error in question is [400 PEER\_ID\_INVALID], and could mean several things:

- The chat id you tried to use is simply wrong, double check it.
- The chat id refers to a group or channel you are not a member of.
- The chat id refers to a user you have't seen yet (from contacts, groups in common, forwarded messages or private chats).
- The chat id argument you passed is in form of a string; you have to convert it into an integer with `int(chat_id)`.

#### 4.31.12 My account has been deactivated/limited!

First of all, you should understand that Telegram wants to be a safe place for people to stay in, and to pursue this goal there are automatic protection systems running to prevent flood and spam, as well as a moderation team of humans who review reports.

Pyrogram is a tool at your commands; it only does what you tell it to do, the rest is up to you.

Having said that, here's a list of what Telegram definitely doesn't like:

- Flood, abusing the API.
- Spam, sending unsolicited messages or adding people to unwanted groups and channels.
- Virtual/VoIP and cheap real numbers, because they are relatively easy to get and likely used for spam/flood.

However, you might be right, and your account was deactivated/limited without any reason. This could happen because of mistakes by either the automatic systems or a moderator. In such cases you can kindly email Telegram at [recover@telegram.org](mailto:recover@telegram.org), contact [@smstelegram](https://twitter.com/smstelegram) on Twitter or use [this form](#).

#### 4.31.13 About the License



Pyrogram is free software and is currently licensed under the terms of the [GNU Lesser General Public License v3 or later \(LGPLv3+\)](#). In short: you may use, redistribute and/or modify it provided that modifications are described and licensed for free under LGPLv3+.

In other words: you can use and integrate Pyrogram into your own code — either open source, under the same or different license, or even proprietary — without being required to release the source code of your own applications. However, any modifications to the library itself are required to be published for free under the same LGPLv3+ license.



## 4.32 Pyrogram Glossary

This page contains a list of common words with brief explanations related to Pyrogram and, to some extent, Telegram in general. Some words may as well link to dedicated articles in case the topic is covered in a more detailed fashion.

---

**Tip:** If you think something interesting could be added here, feel free to propose it by opening a [Feature Request](#).

---

**API** Application Programming Interface: a set of methods, protocols and tools that make it easier to develop programs by providing useful building blocks to the developer.

**API key** A secret code used to authenticate and/or authorize a specific application to Telegram in order for it to control how the API is being used, for example, to prevent abuses of the API. [More on API keys](#).

**DC** Also known as *data center*, is a place where lots of computer systems are housed and used together in order to achieve high quality and availability for services.

**RPC** Acronym for Remote Procedure call, that is, a function which gets executed at some remote place (i.e. Telegram server) and not in your local machine.

**RPCError** An error caused by an RPC which must be returned in place of the successful result in order to let the caller know something went wrong. [More on RPCError](#).

**MTPROTO** The name of the custom-made, open and encrypted protocol by Telegram, implemented in Pyrogram. [More on MTPROTO](#).

**MTPROTO API** The Telegram main API Pyrogram makes use of, which is able to connect both users and normal bots to Telegram using MTPROTO as application layer protocol and execute any method Telegram provides from its public TL-schema. [More on MTPROTO API](#).

**Bot API** The Telegram Bot API that is able to only connect normal bots only to Telegram using HTTP as application layer protocol and allows to execute a sub-set of the main Telegram API. [More on Bot API](#).

**Pyrogrammer** A developer that uses Pyrogram to build Telegram applications.

**Userbot** Also known as *user bot* or *ubot* for short, is a user logged in by third-party Telegram libraries — such as Pyrogram — to automate some behaviours, like sending messages or reacting to text commands or any other event.

**Session** Also known as *login session*, is a strictly personal piece of information created and held by both parties (client and server) which is used to grant permission into a single account without having to start a new authorization process from scratch.

**Callback** Also known as *callback function*, is a user-defined generic function that *can be* registered to and then called-back by the framework when specific events occurs.

**Handler** An object that wraps around a callback function that is *actually meant* to be registered into the framework, which will then be able to handle a specific kind of events, such as a new incoming message, for example. [More on Handlers](#).

**Decorator** Also known as *function decorator*, in Python, is a callable object that is used to modify another function. Decorators in Pyrogram are used to automatically register callback functions for handling updates. [More on Decorators](#).

## 4.33 Release Notes

Release notes for Pyrogram releases will describe what's new in each version, and will also make you aware of any backwards-incompatible changes made in that version.

When upgrading to a new version of Pyrogram, you will need to check all the breaking changes in order to find incompatible code in your application, but also to take advantage of new features and improvements.

---

**Note:** Currently, all Pyrogram release notes live inside the GitHub repository web page: <https://github.com/pyrogram/pyrogram/releases>.

---

## 4.34 Powered by Pyrogram

This is a collection of remarkable projects made with Pyrogram.

---

**Tip:** If you'd like to propose a project that's worth being listed here, feel free to open a [Feature Request](#).

---

### 4.34.1 Projects Showcase

#### YTAudioBot

**A YouTube audio downloader on Telegram, serving over 200k MAU.**

— by [Dan](#)

- Main: <https://t.me/ytaudiobot>
  - Mirror: [https://t.me/ytaudio\\_bot](https://t.me/ytaudio_bot)
  - Website: <https://ytaudiobot.ml>
- 

#### Pyrogram Assistant

**The assistant bot that helps people with Pyrogram directly on Telegram**

— by [Dan](#)

- Bot: <https://t.me/pyrogrambot>
  - Source Code: <https://github.com/pyrogram/assistant>
- 

#### PyroBot

**A Telegram userbot based on Pyrogram**

— by [Colin](#)

- Source Code: <https://git.colinshark.de/PyroBot/PyroBot>
-

## TgIntegration

### Integration Test Library for Telegram Messenger Bots in Python

— by [JosXa](#)

- Source Code: <https://github.com/JosXa/tgintegration>
- 

## BotListBot

### A bot which partly uses Pyrogram to check if other bots are still alive

— by [JosXa](#)

- Source Code: <https://github.com/JosXa/BotListBot>
- 

## 4.35 Support Pyrogram

Pyrogram is free and open source software, and thus supported by your love! If you like the project and have found it to be useful, give Pyrogram a [Star on GitHub](#). Your appreciation means a lot and helps staying motivated.

### 4.35.1 Donate

If you'd also like to donate in order to support Pyrogram – or any of my [other works](#) – you can use the PayPal button below. Thank you.



— Dan

## 4.36 Functions

### 4.36.1 Account

#### AcceptAuthorization

```
class pyrogram.api.functions.account.AcceptAuthorization
```

```
 ID
```

```
 0xe7027c94
```

```
 Parameters
```

- **bot\_id** – int 32-bit
- **scope** – str

- **public\_key** – str
- **value\_hashes** – List of *SecureValueHash*
- **credentials** – *SecureCredentialsEncrypted*

**Returns** bool

### CancelPasswordEmail

**class** pyrogram.api.functions.account.**CancelPasswordEmail**

**ID**

0xc1cbd5b6

No parameters required.

**Returns** bool

### ChangePhone

**class** pyrogram.api.functions.account.**ChangePhone**

**ID**

0x70c32edb

**Parameters**

- **phone\_number** – str
- **phone\_code\_hash** – str
- **phone\_code** – str

**Returns** Either *UserEmpty* or *User*

### CheckUsername

**class** pyrogram.api.functions.account.**CheckUsername**

**ID**

0x2714d86c

**Parameters** **username** – str

**Returns** bool

### ConfirmPasswordEmail

**class** pyrogram.api.functions.account.**ConfirmPasswordEmail**

**ID**

0x8fdf1920

**Parameters** `code` – str

**Returns** bool

### ConfirmPhone

**class** pyrogram.api.functions.account.**ConfirmPhone**

**ID**

0x5f2178c3

**Parameters**

- **phone\_code\_hash** – str
- **phone\_code** – str

**Returns** bool

### DeleteAccount

**class** pyrogram.api.functions.account.**DeleteAccount**

**ID**

0x418d4e0b

**Parameters** **reason** – str

**Returns** bool

### DeleteSecureValue

**class** pyrogram.api.functions.account.**DeleteSecureValue**

**ID**

0xb880bc4b

**Parameters** **types** – List of either *SecureValueTypePersonalDetails*,  
*SecureValueTypePassport*, *SecureValueTypeDriverLicense*,  
*SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*,  
*SecureValueTypeAddress*, *SecureValueTypeUtilityBill*,  
*SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*,  
*SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*,  
*SecureValueTypePhone* or *SecureValueTypeEmail*

**Returns** bool

### FinishTakeoutSession

**class** pyrogram.api.functions.account.**FinishTakeoutSession**

**ID**

0x1d2652ee

**Parameters** **success** (*optional*) – bool

**Returns** bool

### GetAccountTTL

**class** pyrogram.api.functions.account.**GetAccountTTL**

**ID**

0x08fc711d

No parameters required.

**Returns** *AccountDaysTTL*

### GetAllSecureValues

**class** pyrogram.api.functions.account.**GetAllSecureValues**

**ID**

0xb288bc7d

No parameters required.

**Returns** List of *SecureValue*

### GetAuthorizationForm

**class** pyrogram.api.functions.account.**GetAuthorizationForm**

**ID**

0xb86ba8e1

**Parameters**

- **bot\_id** – int 32-bit
- **scope** – str
- **public\_key** – str

**Returns** *account.AuthorizationForm*

### GetAuthorizations

**class** pyrogram.api.functions.account.**GetAuthorizations**

**ID**

0xe320c158

No parameters required.

**Returns** *account.Authorizations*

### GetAutoDownloadSettings

**class** `pyrogram.api.functions.account.GetAutoDownloadSettings`

**ID**

0x56da0b3f

No parameters required.

**Returns** *account.AutoDownloadSettings*

### GetContactSignUpNotification

**class** `pyrogram.api.functions.account.GetContactSignUpNotification`

**ID**

0x9f07c728

No parameters required.

**Returns** `bool`

### GetNotifyExceptions

**class** `pyrogram.api.functions.account.GetNotifyExceptions`

**ID**

0x53577479

**Parameters**

- **compare\_sound** (*optional*) – `bool`
- **peer** (*optional*) – Either *InputNotifyPeer*, *InputNotifyUsers*, *InputNotifyChats* or *InputNotifyBroadcasts*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

### GetNotifySettings

**class** `pyrogram.api.functions.account.GetNotifySettings`

**ID**

0x12b3ad31

**Parameters** **peer** – Either *InputNotifyPeer*, *InputNotifyUsers*, *InputNotifyChats* or *InputNotifyBroadcasts*

**Returns** *PeerNotifySettings*

### GetPassword

**class** pyrogram.api.functions.account.**GetPassword**

**ID**

0x548a30f5

No parameters required.

**Returns** *account.Password*

### GetPasswordSettings

**class** pyrogram.api.functions.account.**GetPasswordSettings**

**ID**

0x9cd4eaf9

**Parameters** **password** – Either *InputCheckPasswordEmpty* or *InputCheckPasswordSRP*

**Returns** *account.PasswordSettings*

### GetPrivacy

**class** pyrogram.api.functions.account.**GetPrivacy**

**ID**

0xdadbc950

**Parameters** **key** – Either *InputPrivacyKeyStatusTimestamp*,  
*InputPrivacyKeyChatInvite*, *InputPrivacyKeyPhoneCall*,  
*InputPrivacyKeyPhoneP2P*, *InputPrivacyKeyForwards* or  
*InputPrivacyKeyProfilePhoto*

**Returns** *account.PrivacyRules*

### GetSecureValue

**class** pyrogram.api.functions.account.**GetSecureValue**

**ID**

0x73665bc2

**Parameters** **types** – List of either *SecureValueTypePersonalDetails*,  
*SecureValueTypePassport*, *SecureValueTypeDriverLicense*,  
*SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*,  
*SecureValueTypeAddress*, *SecureValueTypeUtilityBill*,  
*SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*,  
*SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*,  
*SecureValueTypePhone* or *SecureValueTypeEmail*



**Returns** List of *SecureValue*

### GetTmpPassword

**class** pyrogram.api.functions.account.**GetTmpPassword**

**ID**

0x449e0b51

**Parameters**

- **password** – Either *InputCheckPasswordEmpty* or *InputCheckPasswordSRP*
- **period** – int 32-bit

**Returns** *account.TmpPassword*

### GetWallPaper

**class** pyrogram.api.functions.account.**GetWallPaper**

**ID**

0xfc8ddbea

**Parameters** **wallpaper** – Either *InputWallPaper* or *InputWallPaperSlug*

**Returns** *WallPaper*

### GetWallPapers

**class** pyrogram.api.functions.account.**GetWallPapers**

**ID**

0xaabb1763

**Parameters** **hash** – int 32-bit

**Returns** Either *account.WallPapersNotModified* or *account.WallPapers*

### GetWebAuthorizations

**class** pyrogram.api.functions.account.**GetWebAuthorizations**

**ID**

0x182e6d6f

No parameters required.

**Returns** *account.WebAuthorizations*

## InitTakeoutSession

**class** pyrogram.api.functions.account.**InitTakeoutSession**

**ID**

0xf05b4804

### Parameters

- **contacts** (*optional*) – bool
- **message\_users** (*optional*) – bool
- **message\_chats** (*optional*) – bool
- **message\_megagroups** (*optional*) – bool
- **message\_channels** (*optional*) – bool
- **files** (*optional*) – bool
- **file\_max\_size** (*optional*) – int 32-bit

**Returns** *account.Takeout*

## InstallWallPaper

**class** pyrogram.api.functions.account.**InstallWallPaper**

**ID**

0xfeed5769

### Parameters

- **wallpaper** – Either *InputWallPaper* or *InputWallPaperSlug*
- **settings** – *WallPaperSettings*

**Returns** bool

## RegisterDevice

**class** pyrogram.api.functions.account.**RegisterDevice**

**ID**

0x5cbea590

### Parameters

- **token\_type** – int 32-bit
- **token** – str
- **app\_sandbox** – bool
- **secret** – bytes
- **other\_uids** – List of int 32-bit

**Returns** bool

## ReportPeer

**class** pyrogram.api.functions.account.**ReportPeer**

**ID**

0xae189d5f

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **reason** – Either *InputReportReasonSpam*, *InputReportReasonViolence*, *InputReportReasonPornography*, *InputReportReasonChildAbuse*, *InputReportReasonOther* or *InputReportReasonCopyright*

**Returns** bool

## ResendPasswordEmail

**class** pyrogram.api.functions.account.**ResendPasswordEmail**

**ID**

0x7a7f2a15

No parameters required.

**Returns** bool

## ResetAuthorization

**class** pyrogram.api.functions.account.**ResetAuthorization**

**ID**

0xdf77f3bc

**Parameters** **hash** – int 64-bit

**Returns** bool

## ResetNotifySettings

**class** pyrogram.api.functions.account.**ResetNotifySettings**

**ID**

0xdb7e1747

No parameters required.

**Returns** bool

### ResetWallPapers

**class** pyrogram.api.functions.account.ResetWallPapers

**ID**

0xbb3b9804

No parameters required.

**Returns** bool

### ResetWebAuthorization

**class** pyrogram.api.functions.account.ResetWebAuthorization

**ID**

0x2d01b9ef

**Parameters** **hash** – int 64-bit

**Returns** bool

### ResetWebAuthorizations

**class** pyrogram.api.functions.account.ResetWebAuthorizations

**ID**

0x682d2594

No parameters required.

**Returns** bool

### SaveAutoDownloadSettings

**class** pyrogram.api.functions.account.SaveAutoDownloadSettings

**ID**

0x76f36233

**Parameters**

- **settings** – *AutoDownloadSettings*
- **low**(*optional*) – bool
- **high**(*optional*) – bool

**Returns** bool

### SaveSecureValue

**class** pyrogram.api.functions.account.**SaveSecureValue**

**ID**

0x899fe31d

**Parameters**

- **value** – *InputSecureValue*
- **secure\_secret\_id** – int 64-bit

**Returns** *SecureValue*

### SaveWallPaper

**class** pyrogram.api.functions.account.**SaveWallPaper**

**ID**

0x6c5a5b37

**Parameters**

- **wallpaper** – Either *InputWallPaper* or *InputWallPaperSlug*
- **unsave** – bool
- **settings** – *WallPaperSettings*

**Returns** bool

### SendChangePhoneCode

**class** pyrogram.api.functions.account.**SendChangePhoneCode**

**ID**

0x82574ae5

**Parameters**

- **phone\_number** – str
- **settings** – *CodeSettings*

**Returns** *auth.SentCode*

### SendConfirmPhoneCode

**class** pyrogram.api.functions.account.**SendConfirmPhoneCode**

**ID**

0x1b3faa88

**Parameters**

- **hash** – str
- **settings** – *CodeSettings*

**Returns** *auth.SentCode*

### SendVerifyEmailCode

**class** pyrogram.api.functions.account.**SendVerifyEmailCode**

**ID**

0x7011509f

**Parameters** **email** – str

**Returns** *account.SentEmailCode*

### SendVerifyPhoneCode

**class** pyrogram.api.functions.account.**SendVerifyPhoneCode**

**ID**

0xa5a356f9

**Parameters**

- **phone\_number** – str
- **settings** – *CodeSettings*

**Returns** *auth.SentCode*

### SetAccountTTL

**class** pyrogram.api.functions.account.**SetAccountTTL**

**ID**

0x2442485e

**Parameters** **ttl** – *AccountDaysTTL*

**Returns** bool

### SetContactSignUpNotification

**class** pyrogram.api.functions.account.**SetContactSignUpNotification**

**ID**

0xcff43f61

**Parameters** **silent** – bool

**Returns** bool

## SetPrivacy

```
class pyrogram.api.functions.account.SetPrivacy
```

**ID**

0xc9f81ce8

### Parameters

- **key** – Either *InputPrivacyKeyStatusTimestamp*,  
*InputPrivacyKeyChatInvite*, *InputPrivacyKeyPhoneCall*,  
*InputPrivacyKeyPhoneP2P*, *InputPrivacyKeyForwards* or  
*InputPrivacyKeyProfilePhoto*
- **rules** – List of either *InputPrivacyValueAllowContacts*,  
*InputPrivacyValueAllowAll*, *InputPrivacyValueAllowUsers*,  
*InputPrivacyValueDisallowContacts*, *InputPrivacyValueDisallowAll*  
or *InputPrivacyValueDisallowUsers*

**Returns** *account.PrivacyRules*

## UnregisterDevice

```
class pyrogram.api.functions.account.UnregisterDevice
```

**ID**

0x3076c4bf

### Parameters

- **token\_type** – int 32-bit
- **token** – str
- **other\_uids** – List of int 32-bit

**Returns** bool

## UpdateDeviceLocked

```
class pyrogram.api.functions.account.UpdateDeviceLocked
```

**ID**

0x38df3532

**Parameters** **period** – int 32-bit

**Returns** bool

## UpdateNotifySettings

```
class pyrogram.api.functions.account.UpdateNotifySettings
```

**ID**

0x84be5b93

**Parameters**

- **peer** – Either *InputNotifyPeer*, *InputNotifyUsers*, *InputNotifyChats* or *InputNotifyBroadcasts*
- **settings** – *InputPeerNotifySettings*

**Returns** bool

## UpdatePasswordSettings

```
class pyrogram.api.functions.account.UpdatePasswordSettings
```

**ID**

0xa59b102f

**Parameters**

- **password** – Either *InputCheckPasswordEmpty* or *InputCheckPasswordSRP*
- **new\_settings** – *account.PasswordInputSettings*

**Returns** bool

## UpdateProfile

```
class pyrogram.api.functions.account.UpdateProfile
```

**ID**

0x78515775

**Parameters**

- **first\_name** (*optional*) – str
- **last\_name** (*optional*) – str
- **about** (*optional*) – str

**Returns** Either *UserEmpty* or *User*

## UpdateStatus

```
class pyrogram.api.functions.account.UpdateStatus
```

**ID**

0x6628562c

**Parameters** **offline** – bool

**Returns** bool



### UpdateUsername

```
class pyrogram.api.functions.account.UpdateUsername
```

**ID**

0x3e0bdd7c

**Parameters** `username` – str

**Returns** Either *UserEmpty* or *User*

### UploadWallPaper

```
class pyrogram.api.functions.account.UploadWallPaper
```

**ID**

0xdd853661

**Parameters**

- **file** – Either *InputFile* or *InputFileBig*
- **mime\_type** – str
- **settings** – *WallPaperSettings*

**Returns** *WallPaper*

### VerifyEmail

```
class pyrogram.api.functions.account.VerifyEmail
```

**ID**

0xecba39db

**Parameters**

- **email** – str
- **code** – str

**Returns** bool

### VerifyPhone

```
class pyrogram.api.functions.account.VerifyPhone
```

**ID**

0x4dd3a7f6

**Parameters**

- **phone\_number** – str
- **phone\_code\_hash** – str

- **phone\_code** – str

**Returns** bool

## 4.36.2 Auth

### BindTempAuthKey

**class** pyrogram.api.functions.auth.**BindTempAuthKey**

**ID**

0xcdd42a05

**Parameters**

- **perm\_auth\_key\_id** – int 64-bit
- **nonce** – int 64-bit
- **expires\_at** – int 32-bit
- **encrypted\_message** – bytes

**Returns** bool

### CancelCode

**class** pyrogram.api.functions.auth.**CancelCode**

**ID**

0x1f040578

**Parameters**

- **phone\_number** – str
- **phone\_code\_hash** – str

**Returns** bool

### CheckPassword

**class** pyrogram.api.functions.auth.**CheckPassword**

**ID**

0xd18b4d16

**Parameters** **password** – Either *InputCheckPasswordEmpty* or *InputCheckPasswordSRP*

**Returns** *auth.Authorization*

### DropTempAuthKeys

**class** pyrogram.api.functions.auth.DropTempAuthKeys

**ID**

0x8e48a188

**Parameters** **except\_auth\_keys** – List of int 64-bit

**Returns** bool

### ExportAuthorization

**class** pyrogram.api.functions.auth.ExportAuthorization

**ID**

0xe5bffffcd

**Parameters** **dc\_id** – int 32-bit

**Returns** *auth.ExportedAuthorization*

### ImportAuthorization

**class** pyrogram.api.functions.auth.ImportAuthorization

**ID**

0xe3ef9613

**Parameters**

- **id** – int 32-bit
- **bytes** – bytes

**Returns** *auth.Authorization*

### ImportBotAuthorization

**class** pyrogram.api.functions.auth.ImportBotAuthorization

**ID**

0x67a3ff2c

**Parameters**

- **flags** – int 32-bit
- **api\_id** – int 32-bit
- **api\_hash** – str
- **bot\_auth\_token** – str

**Returns** *auth.Authorization*

## LogOut

**class** pyrogram.api.functions.auth.LogOut

**ID**

0x5717da40

No parameters required.

**Returns** bool

## RecoverPassword

**class** pyrogram.api.functions.auth.RecoverPassword

**ID**

0x4ea56e92

**Parameters** `code` – str

**Returns** *auth.Authorization*

## RequestPasswordRecovery

**class** pyrogram.api.functions.auth.RequestPasswordRecovery

**ID**

0xd897bc66

No parameters required.

**Returns** *auth.PasswordRecovery*

## ResendCode

**class** pyrogram.api.functions.auth.ResendCode

**ID**

0x3ef1a9bf

**Parameters**

- `phone_number` – str
- `phone_code_hash` – str

**Returns** *auth.SentCode*

## ResetAuthorizations

**class** pyrogram.api.functions.auth.ResetAuthorizations

**ID**

0x9fab0d1a

No parameters required.

**Returns** `bool`**SendCode****class** `pyrogram.api.functions.auth.SendCode`**ID**

0xa677244f

**Parameters**

- **phone\_number** – `str`
- **api\_id** – `int` 32-bit
- **api\_hash** – `str`
- **settings** – `CodeSettings`

**Returns** `auth.SentCode`**SignIn****class** `pyrogram.api.functions.auth.SignIn`**ID**

0xbcd51581

**Parameters**

- **phone\_number** – `str`
- **phone\_code\_hash** – `str`
- **phone\_code** – `str`

**Returns** `auth.Authorization`**SignUp****class** `pyrogram.api.functions.auth.SignUp`**ID**

0x1b067634

**Parameters**

- **phone\_number** – `str`
- **phone\_code\_hash** – `str`
- **phone\_code** – `str`

- **first\_name** – str
- **last\_name** – str

**Returns** *auth.Authorization*

### 4.36.3 Bots

#### AnswerWebhookJSONQuery

**class** pyrogram.api.functions.bots.**AnswerWebhookJSONQuery**

**ID**

0xe6213f4d

##### Parameters

- **query\_id** – int 64-bit
- **data** – *DataJSON*

**Returns** bool

#### SendCustomRequest

**class** pyrogram.api.functions.bots.**SendCustomRequest**

**ID**

0xaa2769ed

##### Parameters

- **custom\_method** – str
- **params** – *DataJSON*

**Returns** *DataJSON*

### 4.36.4 Channels

#### CheckUsername

**class** pyrogram.api.functions.channels.**CheckUsername**

**ID**

0x10e6bd2c

##### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **username** – str

**Returns** bool

## CreateChannel

```
class pyrogram.api.functions.channels.CreateChannel
```

**ID**

0xf4893d7f

### Parameters

- **title** – str
- **about** – str
- **broadcast** (*optional*) – bool
- **megagroup** (*optional*) – bool

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## DeleteChannel

```
class pyrogram.api.functions.channels.DeleteChannel
```

**ID**

0xc0111fe3

**Parameters** **channel** – Either *InputChannelEmpty* or *InputChannel*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## DeleteHistory

```
class pyrogram.api.functions.channels.DeleteHistory
```

**ID**

0xaf369d42

### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **max\_id** – int 32-bit

**Returns** bool

## DeleteMessages

```
class pyrogram.api.functions.channels.DeleteMessages
```

**ID**

0x84c1fd4e

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **id** – List of int 32-bit

**Returns** *messages.AffectedMessages***DeleteUserHistory****class** pyrogram.api.functions.channels.**DeleteUserHistory****ID**

0xd10dd71b

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** *messages.AffectedHistory***EditAdmin****class** pyrogram.api.functions.channels.**EditAdmin****ID**

0x70f893ba

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **admin\_rights** – *ChatAdminRights*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage***EditBanned****class** pyrogram.api.functions.channels.**EditBanned****ID**

0x72796912

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*



- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **banned\_rights** – *ChatBannedRights*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## EditPhoto

**class** pyrogram.api.functions.channels.**EditPhoto**

**ID**

0xf12e57c9

### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **photo** – Either *InputChatPhotoEmpty*, *InputChatUploadedPhoto* or *InputChatPhoto*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## EditTitle

**class** pyrogram.api.functions.channels.**EditTitle**

**ID**

0x566decd0

### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **title** – *str*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## ExportInvite

**class** pyrogram.api.functions.channels.**ExportInvite**

**ID**

0xc7560885

**Parameters** **channel** – Either *InputChannelEmpty* or *InputChannel*

**Returns** Either *ChatInviteEmpty* or *ChatInviteExported*

## ExportMessageLink

**class** pyrogram.api.functions.channels.**ExportMessageLink**

**ID**

0xceb77163

### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **id** – int 32-bit
- **grouped** – bool

**Returns** *ExportedMessageLink*

## GetAdminLog

**class** pyrogram.api.functions.channels.**GetAdminLog**

**ID**

0x33ddf480

### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **q** – str
- **max\_id** – int 64-bit
- **min\_id** – int 64-bit
- **limit** – int 32-bit
- **events\_filter** (optional) – *ChannelAdminLogEventsFilter*
- **admins** (optional) – List of either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** *channels.AdminLogResults*

## GetAdmindedPublicChannels

**class** pyrogram.api.functions.channels.**GetAdmindedPublicChannels**

**ID**

0x8d8d82d7

No parameters required.

**Returns** Either *messages.Chats* or *messages.ChatsSlice*

### GetChannels

**class** pyrogram.api.functions.channels.**GetChannels**

**ID**

0x0a7f6bbb

**Parameters** **id** – List of either *InputChannelEmpty* or *InputChannel*

**Returns** Either *messages.Chats* or *messages.ChatsSlice*

### GetFullChannel

**class** pyrogram.api.functions.channels.**GetFullChannel**

**ID**

0x08736a09

**Parameters** **channel** – Either *InputChannelEmpty* or *InputChannel*

**Returns** *messages.ChatFull*

### GetLeftChannels

**class** pyrogram.api.functions.channels.**GetLeftChannels**

**ID**

0x8341ecc0

**Parameters** **offset** – int 32-bit

**Returns** Either *messages.Chats* or *messages.ChatsSlice*

### GetMessages

**class** pyrogram.api.functions.channels.**GetMessages**

**ID**

0xad8c9a23

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **id** – List of either *InputMessageID*, *InputMessageReplyTo* or *InputMessagePinned*

**Returns** Either *messages.Messages*, *messages.MessagesSlice*, *messages.ChannelMessages* or *messages.MessagesNotModified*

## GetParticipant

**class** pyrogram.api.functions.channels.**GetParticipant**

**ID**

0x546dd7a6

### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** *channels.ChannelParticipant*

## GetParticipants

**class** pyrogram.api.functions.channels.**GetParticipants**

**ID**

0x123e05e9

### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **filter** – Either *ChannelParticipantsRecent*, *ChannelParticipantsAdmins*, *ChannelParticipantsKicked*, *ChannelParticipantsBots*, *ChannelParticipantsBanned*, *ChannelParticipantsSearch* or *ChannelParticipantsContacts*
- **offset** – int 32-bit
- **limit** – int 32-bit
- **hash** – int 32-bit

**Returns** Either *channels.ChannelParticipants* or *channels.ChannelParticipantsNotModified*

## InviteToChannel

**class** pyrogram.api.functions.channels.**InviteToChannel**

**ID**

0x199f3a6c

### Parameters

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **users** – List of either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## JoinChannel

```
class pyrogram.api.functions.channels.JoinChannel
```

**ID**

0x24b524c5

**Parameters** **channel** – Either *InputChannelEmpty* or *InputChannel*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## LeaveChannel

```
class pyrogram.api.functions.channels.LeaveChannel
```

**ID**

0xf836aa95

**Parameters** **channel** – Either *InputChannelEmpty* or *InputChannel*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## ReadHistory

```
class pyrogram.api.functions.channels.ReadHistory
```

**ID**

0xcc104937

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **max\_id** – int 32-bit

**Returns** bool

## ReadMessageContents

```
class pyrogram.api.functions.channels.ReadMessageContents
```

**ID**

0xeab5dc38

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **id** – List of int 32-bit

**Returns** bool

## ReportSpam

**class** pyrogram.api.functions.channels.**ReportSpam**

**ID**

0xfe087810

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **id** – List of int 32-bit

**Returns** bool

## SetStickers

**class** pyrogram.api.functions.channels.**SetStickers**

**ID**

0xea8ca4f9

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **stickerset** – Either *InputStickerSetEmpty*, *InputStickerSetID* or *InputStickerSetShortName*

**Returns** bool

## TogglePreHistoryHidden

**class** pyrogram.api.functions.channels.**TogglePreHistoryHidden**

**ID**

0xeabbb94c

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **enabled** – bool

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## ToggleSignatures

**class** pyrogram.api.functions.channels.ToggleSignatures

**ID**

0x1f69b606

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **enabled** – bool

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## UpdateUsername

**class** pyrogram.api.functions.channels.UpdateUsername

**ID**

0x3514b3de

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **username** – str

**Returns** bool

## 4.36.5 Contacts

### Block

**class** pyrogram.api.functions.contacts.Block

**ID**

0x332b49fc

**Parameters** **id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** bool

### DeleteByPhones

**class** pyrogram.api.functions.contacts.DeleteByPhones

**ID**

0x1013fd9e

**Parameters** **phones** – List of str

**Returns** bool

### DeleteContact

**class** pyrogram.api.functions.contacts.DeleteContact

**ID**

0x8e953744

**Parameters** *id* – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** *contacts.Link*

### DeleteContacts

**class** pyrogram.api.functions.contacts.DeleteContacts

**ID**

0x59ab389e

**Parameters** *id* – List of either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** bool

### GetBlocked

**class** pyrogram.api.functions.contacts.GetBlocked

**ID**

0xf57c350f

**Parameters**

- **offset** – int 32-bit
- **limit** – int 32-bit

**Returns** Either *contacts.Blocked* or *contacts.BlockedSlice*

### GetContactIDs

**class** pyrogram.api.functions.contacts.GetContactIDs

**ID**

0x2caa4a42

**Parameters** *hash* – int 32-bit

**Returns** List of int 32-bit



### GetContacts

**class** pyrogram.api.functions.contacts.**GetContacts**

**ID**

0xc023849f

**Parameters** **hash** – int 32-bit

**Returns** Either *contacts.ContactsNotModified* or *contacts.Contacts*

### GetSaved

**class** pyrogram.api.functions.contacts.**GetSaved**

**ID**

0x82f1e39f

No parameters required.

**Returns** List of *SavedPhoneContact*

### GetStatuses

**class** pyrogram.api.functions.contacts.**GetStatuses**

**ID**

0xc4a353ee

No parameters required.

**Returns** List of *ContactStatus*

### GetTopPeers

**class** pyrogram.api.functions.contacts.**GetTopPeers**

**ID**

0xd4982db5

**Parameters**

- **offset** – int 32-bit
- **limit** – int 32-bit
- **hash** – int 32-bit
- **correspondents** (*optional*) – bool
- **bots\_pm** (*optional*) – bool
- **bots\_inline** (*optional*) – bool
- **phone\_calls** (*optional*) – bool

- **groups** (*optional*) – bool
- **channels** (*optional*) – bool

**Returns** Either `contacts.TopPeersNotModified`, `contacts.TopPeers` or `contacts.TopPeersDisabled`

## ImportContacts

**class** pyrogram.api.functions.contacts.**ImportContacts**

**ID**

0x2c800be5

**Parameters** **contacts** – List of *InputPhoneContact*

**Returns** *contacts.ImportedContacts*

## ResetSaved

**class** pyrogram.api.functions.contacts.**ResetSaved**

**ID**

0x879537f1

No parameters required.

**Returns** bool

## ResetTopPeerRating

**class** pyrogram.api.functions.contacts.**ResetTopPeerRating**

**ID**

0x1ae373ac

**Parameters**

- **category** – Either *TopPeerCategoryBotsPM*, *TopPeerCategoryBotsInline*, *TopPeerCategoryCorrespondents*, *TopPeerCategoryGroups*, *TopPeerCategoryChannels* or *TopPeerCategoryPhoneCalls*
- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*

**Returns** bool

## ResolveUsername

**class** pyrogram.api.functions.contacts.**ResolveUsername**

**ID**

0xf93ccba3

**Parameters** `username` – str

**Returns** `contacts.ResolvedPeer`

## Search

**class** `pyrogram.api.functions.contacts.Search`

**ID**

`0x11f812d8`

**Parameters**

- `q` – str
- `limit` – int 32-bit

**Returns** `contacts.Found`

## ToggleTopPeers

**class** `pyrogram.api.functions.contacts.ToggleTopPeers`

**ID**

`0x8514bdda`

**Parameters** `enabled` – bool

**Returns** bool

## Unblock

**class** `pyrogram.api.functions.contacts.Unblock`

**ID**

`0xe54100bd`

**Parameters** `id` – Either `InputUserEmpty`, `InputUserSelf` or `InputUser`

**Returns** bool

## 4.36.6 Contest

### SaveDeveloperInfo

**class** `pyrogram.api.functions.contest.SaveDeveloperInfo`

**ID**

`0x9a5f6e95`

**Parameters**

- **vk\_id** – int 32-bit
- **name** – str
- **phone\_number** – str
- **age** – int 32-bit
- **city** – str

**Returns** bool

### 4.36.7 Help

#### AcceptTermsOfService

**class** pyrogram.api.functions.help.**AcceptTermsOfService**

**ID**

0xee72f79a

**Parameters** **id** – *DataJSON*

**Returns** bool

#### EditUserInfo

**class** pyrogram.api.functions.help.**EditUserInfo**

**ID**

0x66b91b70

**Parameters**

- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **message** – str
- **entities** – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

**Returns** Either *help.UserInfoEmpty* or *help.UserInfo*

#### GetAppChangelog

**class** pyrogram.api.functions.help.**GetAppChangelog**

**ID**

0x9010ef6f

**Parameters** `prev_app_version` – str

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

### GetAppConfig

**class** `pyrogram.api.functions.help.GetAppConfig`

**ID**

0x98914110

No parameters required.

**Returns** Either *JsonNull*, *JsonBool*, *JsonNumber*, *JsonString*, *JsonArray* or  
*JsonObject*

### GetAppUpdate

**class** `pyrogram.api.functions.help.GetAppUpdate`

**ID**

0x522d5a7d

**Parameters** `source` – str

**Returns** Either *help.AppUpdate* or *help.NoAppUpdate*

### GetCdnConfig

**class** `pyrogram.api.functions.help.GetCdnConfig`

**ID**

0x52029342

No parameters required.

**Returns** *CdnConfig*

### GetConfig

**class** `pyrogram.api.functions.help.GetConfig`

**ID**

0xc4f9186b

No parameters required.

**Returns** *Config*

### GetDeepLinkInfo

**class** pyrogram.api.functions.help.GetDeepLinkInfo

**ID**

0x3fedc75f

**Parameters** *path* – str

**Returns** Either *help.DeepLinkInfoEmpty* or *help.DeepLinkInfo*

### GetInviteText

**class** pyrogram.api.functions.help.GetInviteText

**ID**

0x4d392343

No parameters required.

**Returns** *help.InviteText*

### GetNearestDc

**class** pyrogram.api.functions.help.GetNearestDc

**ID**

0x1fb33026

No parameters required.

**Returns** *NearestDc*

### GetPassportConfig

**class** pyrogram.api.functions.help.GetPassportConfig

**ID**

0xc661ad08

**Parameters** *hash* – int 32-bit

**Returns** Either *help.PassportConfigNotModified* or *help.PassportConfig*

### GetProxyData

**class** pyrogram.api.functions.help.GetProxyData

**ID**

0x3d7758e1

No parameters required.

**Returns** Either *help.ProxyDataEmpty* or *help.ProxyDataPromo*

### GetRecentMeUrls

**class** pyrogram.api.functions.help.GetRecentMeUrls

**ID**

0x3dc0f114

**Parameters** **referer** – str

**Returns** *help.RecentMeUrls*

### GetSupport

**class** pyrogram.api.functions.help.GetSupport

**ID**

0x9cdf08cd

No parameters required.

**Returns** *help.Support*

### GetSupportName

**class** pyrogram.api.functions.help.GetSupportName

**ID**

0xd360e72c

No parameters required.

**Returns** *help.SupportName*

### GetTermsOfServiceUpdate

**class** pyrogram.api.functions.help.GetTermsOfServiceUpdate

**ID**

0x2ca51fd1

No parameters required.

**Returns** Either *help.TermsOfServiceUpdateEmpty* or *help.TermsOfServiceUpdate*

### GetUserInfo

**class** pyrogram.api.functions.help.**GetUserInfo**

**ID**

0x038a08d3

**Parameters** **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** Either *help.UserInfoEmpty* or *help.UserInfo*

### SaveAppLog

**class** pyrogram.api.functions.help.**SaveAppLog**

**ID**

0x6f02f748

**Parameters** **events** – List of *InputAppEvent*

**Returns** bool

### SetBotUpdatesStatus

**class** pyrogram.api.functions.help.**SetBotUpdatesStatus**

**ID**

0xec22cfcd

**Parameters**

- **pending\_updates\_count** – int 32-bit
- **message** – str

**Returns** bool

## 4.36.8 Langpack

### GetDifference

**class** pyrogram.api.functions.langpack.**GetDifference**

**ID**

0xcd984aa5

**Parameters**

- **lang\_pack** – str
- **lang\_code** – str
- **from\_version** – int 32-bit



**Returns** *LangPackDifference*

### GetLangPack

**class** pyrogram.api.functions.langpack.**GetLangPack**

**ID**  
0xf2f2330a

**Parameters**

- **lang\_pack** – str
- **lang\_code** – str

**Returns** *LangPackDifference*

### GetLanguage

**class** pyrogram.api.functions.langpack.**GetLanguage**

**ID**  
0x6a596502

**Parameters**

- **lang\_pack** – str
- **lang\_code** – str

**Returns** *LangPackLanguage*

### GetLanguages

**class** pyrogram.api.functions.langpack.**GetLanguages**

**ID**  
0x42c6978f

**Parameters** **lang\_pack** – str

**Returns** List of *LangPackLanguage*

### GetStrings

**class** pyrogram.api.functions.langpack.**GetStrings**

**ID**  
0xefea3803

**Parameters**

- **lang\_pack** – str

- **lang\_code** – str
- **keys** – List of str

**Returns** List of either *LangPackString*, *LangPackStringPluralized* or *LangPackStringDeleted*

## 4.36.9 Messages

### AcceptEncryption

**class** pyrogram.api.functions.messages.**AcceptEncryption**

**ID**

0x3dbc0415

#### Parameters

- **peer** – *InputEncryptedChat*
- **g\_b** – bytes
- **key\_fingerprint** – int 64-bit

**Returns** Either *EncryptedChatEmpty*, *EncryptedChatWaiting*, *EncryptedChatRequested*, *EncryptedChat* or *EncryptedChatDiscarded*

### AddChatUser

**class** pyrogram.api.functions.messages.**AddChatUser**

**ID**

0xf9a0aa09

#### Parameters

- **chat\_id** – int 32-bit
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **fwd\_limit** – int 32-bit

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

### CheckChatInvite

**class** pyrogram.api.functions.messages.**CheckChatInvite**

**ID**

0x3eadb1bb

**Parameters** **hash** – str

**Returns** Either *ChatInviteAlready* or *ChatInvite*

### ClearAllDrafts

```
class pyrogram.api.functions.messages.ClearAllDrafts
```

**ID**

0x7e58ee9c

No parameters required.

**Returns** bool

### ClearRecentStickers

```
class pyrogram.api.functions.messages.ClearRecentStickers
```

**ID**

0x8999602d

**Parameters** **attached** (*optional*) – bool

**Returns** bool

### CreateChat

```
class pyrogram.api.functions.messages.CreateChat
```

**ID**

0x09cb126e

**Parameters**

- **users** – List of either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **title** – str

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

### DeleteChatUser

```
class pyrogram.api.functions.messages.DeleteChatUser
```

**ID**

0xe0611f16

**Parameters**

- **chat\_id** – int 32-bit
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## DeleteHistory

**class** pyrogram.api.functions.messages.DeleteHistory

**ID**

0x1c015b09

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*,  
*InputPeerUser* or *InputPeerChannel*
- **max\_id** – int 32-bit
- **just\_clear** (*optional*) – bool
- **revoke** (*optional*) – bool

**Returns** *messages.AffectedHistory*

## DeleteMessages

**class** pyrogram.api.functions.messages.DeleteMessages

**ID**

0xe58e95d2

**Parameters**

- **id** – List of int 32-bit
- **revoke** (*optional*) – bool

**Returns** *messages.AffectedMessages*

## DiscardEncryption

**class** pyrogram.api.functions.messages.DiscardEncryption

**ID**

0xedd923c5

**Parameters** **chat\_id** – int 32-bit

**Returns** bool

### EditChatAbout

```
class pyrogram.api.functions.messages.EditChatAbout
```

**ID**

0xdef60797

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **about** – str

**Returns** bool

### EditChatAdmin

```
class pyrogram.api.functions.messages.EditChatAdmin
```

**ID**

0xa9e69f2e

**Parameters**

- **chat\_id** – int 32-bit
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **is\_admin** – bool

**Returns** bool

### EditChatDefaultBannedRights

```
class pyrogram.api.functions.messages.EditChatDefaultBannedRights
```

**ID**

0xa5866b41

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **banned\_rights** – *ChatBannedRights*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## EditChatPhoto

**class** pyrogram.api.functions.messages.**EditChatPhoto**

**ID**

0xca4c79d8

### Parameters

- **chat\_id** – int 32-bit
- **photo** – Either *InputChatPhotoEmpty*, *InputChatUploadedPhoto* or *InputChatPhoto*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## EditChatTitle

**class** pyrogram.api.functions.messages.**EditChatTitle**

**ID**

0xdc452855

### Parameters

- **chat\_id** – int 32-bit
- **title** – str

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## EditInlineBotMessage

**class** pyrogram.api.functions.messages.**EditInlineBotMessage**

**ID**

0x83557dba

### Parameters

- **id** – *InputBotInlineMessageID*
- **no\_webpage** (optional) – bool
- **message** (optional) – str
- **media** (optional) – Either *InputMediaEmpty*,  
*InputMediaUploadedPhoto*, *InputMediaPhoto*, *InputMediaGeoPoint*,  
*InputMediaContact*, *InputMediaUploadedDocument*,  
*InputMediaDocument*, *InputMediaVenue*, *InputMediaGifExternal*,  
*InputMediaPhotoExternal*, *InputMediaDocumentExternal*,

*InputMediaGame, InputMediaInvoice, InputMediaGeoLive or  
InputMediaPoll*

- **reply\_markup** (optional) – Either *ReplyKeyboardHide, ReplyKeyboardForceReply, ReplyKeyboardMarkup or ReplyInlineMarkup*
- **entities** (optional) – List of either *MessageEntityUnknown, MessageEntityMention, MessageEntityHashtag, MessageEntityBotCommand, MessageEntityUrl, MessageEntityEmail, MessageEntityBold, MessageEntityItalic, MessageEntityCode, MessageEntityPre, MessageEntityTextUrl, MessageEntityMentionName, InputMessageEntityMentionName, MessageEntityPhone or MessageEntityCashtag*

**Returns** bool

## EditMessage

**class** pyrogram.api.functions.messages.**EditMessage**

**ID**

0xd116f31e

### Parameters

- **peer** – Either *InputPeerEmpty, InputPeerSelf, InputPeerChat, InputPeerUser or InputPeerChannel*
- **id** – int 32-bit
- **no\_webpage** (optional) – bool
- **message** (optional) – str
- **media** (optional) – Either *InputMediaEmpty, InputMediaUploadedPhoto, InputMediaPhoto, InputMediaGeoPoint, InputMediaContact, InputMediaUploadedDocument, InputMediaDocument, InputMediaVenue, InputMediaGifExternal, InputMediaPhotoExternal, InputMediaDocumentExternal, InputMediaGame, InputMediaInvoice, InputMediaGeoLive or InputMediaPoll*
- **reply\_markup** (optional) – Either *ReplyKeyboardHide, ReplyKeyboardForceReply, ReplyKeyboardMarkup or ReplyInlineMarkup*
- **entities** (optional) – List of either *MessageEntityUnknown, MessageEntityMention, MessageEntityHashtag, MessageEntityBotCommand, MessageEntityUrl, MessageEntityEmail, MessageEntityBold, MessageEntityItalic, MessageEntityCode, MessageEntityPre, MessageEntityTextUrl, MessageEntityMentionName, InputMessageEntityMentionName, MessageEntityPhone or MessageEntityCashtag*

**Returns** Either *UpdatesTooLong, UpdateShortMessage, UpdateShortChatMessage, UpdateShort, UpdatesCombined, Update or UpdateShortSentMessage*

## ExportChatInvite

**class** pyrogram.api.functions.messages.**ExportChatInvite**

**ID**

0x0df7534c

**Parameters** **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*

**Returns** Either *ChatInviteEmpty* or *ChatInviteExported*

## FaveSticker

**class** pyrogram.api.functions.messages.**FaveSticker**

**ID**

0xb9ffc55b

**Parameters**

- **id** – Either *InputDocumentEmpty* or *InputDocument*
- **unfave** – bool

**Returns** bool

## ForwardMessages

**class** pyrogram.api.functions.messages.**ForwardMessages**

**ID**

0x708e0195

**Parameters**

- **from\_peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **id** – List of int 32-bit
- **random\_id** – List of int 64-bit
- **to\_peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **silent** (*optional*) – bool
- **background** (*optional*) – bool
- **with\_my\_score** (*optional*) – bool
- **grouped** (*optional*) – bool

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*



### GetAllChats

**class** pyrogram.api.functions.messages.GetAllChats

**ID**

0xeba80ff0

**Parameters** **except\_ids** – List of int 32-bit

**Returns** Either *messages.Chats* or *messages.ChatsSlice*

### GetAllDrafts

**class** pyrogram.api.functions.messages.GetAllDrafts

**ID**

0x6a3f8d65

No parameters required.

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

### GetAllStickers

**class** pyrogram.api.functions.messages.GetAllStickers

**ID**

0x1c9618b1

**Parameters** **hash** – int 32-bit

**Returns** Either *messages.AllStickersNotModified* or *messages.AllStickers*

### GetArchivedStickers

**class** pyrogram.api.functions.messages.GetArchivedStickers

**ID**

0x57f17692

**Parameters**

- **offset\_id** – int 64-bit
- **limit** – int 32-bit
- **masks** (*optional*) – bool

**Returns** *messages.ArchivedStickers*

### GetAttachedStickers

**class** pyrogram.api.functions.messages.**GetAttachedStickers**

**ID**

0xcc5b67cc

**Parameters** **media** – Either *InputStickeredMediaPhoto* or *InputStickeredMediaDocument*

**Returns** List of either *StickerSetCovered* or *StickerSetMultiCovered*

### GetBotCallbackAnswer

**class** pyrogram.api.functions.messages.**GetBotCallbackAnswer**

**ID**

0x810a9fec

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **msg\_id** – int 32-bit
- **game** (*optional*) – bool
- **data** (*optional*) – bytes

**Returns** *messages.BotCallbackAnswer*

### GetChats

**class** pyrogram.api.functions.messages.**GetChats**

**ID**

0x3c6aa187

**Parameters** **id** – List of int 32-bit

**Returns** Either *messages.Chats* or *messages.ChatsSlice*

### GetCommonChats

**class** pyrogram.api.functions.messages.**GetCommonChats**

**ID**

0x0d0a48c4

**Parameters**

- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

- **max\_id** – int 32-bit
- **limit** – int 32-bit

**Returns** Either *messages.Chats* or *messages.ChatsSlice*

### GetDhConfig

**class** pyrogram.api.functions.messages.GetDhConfig

**ID**

0x26cf8950

#### Parameters

- **version** – int 32-bit
- **random\_length** – int 32-bit

**Returns** Either *messages.DhConfigNotModified* or *messages.DhConfig*

### GetDialogUnreadMarks

**class** pyrogram.api.functions.messages.GetDialogUnreadMarks

**ID**

0x22e24e22

No parameters required.

**Returns** List of *DialogPeer*

### GetDialogs

**class** pyrogram.api.functions.messages.GetDialogs

**ID**

0xb098aee6

#### Parameters

- **offset\_date** – int 32-bit
- **offset\_id** – int 32-bit
- **offset\_peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **limit** – int 32-bit
- **hash** – int 32-bit
- **exclude\_pinned** (*optional*) – bool

**Returns** Either *messages.Dialogs*, *messages.DialogsSlice* or *messages.DialogsNotModified*

### GetDocumentByHash

```
class pyrogram.api.functions.messages.GetDocumentByHash
```

**ID**

0x338e2464

**Parameters**

- **sha256** – bytes
- **size** – int 32-bit
- **mime\_type** – str

**Returns** Either *DocumentEmpty* or *Document*

### GetEmojiKeywords

```
class pyrogram.api.functions.messages.GetEmojiKeywords
```

**ID**

0x35a0e062

**Parameters** **lang\_code** – str

**Returns** *EmojiKeywordsDifference*

### GetEmojiKeywordsDifference

```
class pyrogram.api.functions.messages.GetEmojiKeywordsDifference
```

**ID**

0x1508b6af

**Parameters**

- **lang\_code** – str
- **from\_version** – int 32-bit

**Returns** *EmojiKeywordsDifference*

### GetEmojiURL

```
class pyrogram.api.functions.messages.GetEmojiURL
```

**ID**

0xd5b10c26

**Parameters** **lang\_code** – str

**Returns** *EmojiURL*

### GetFavedStickers

**class** pyrogram.api.functions.messages.**GetFavedStickers**

**ID**

0x21ce0b0e

**Parameters** **hash** – int 32-bit

**Returns** Either *messages.FavedStickersNotModified* or *messages.FavedStickers*

### GetFeaturedStickers

**class** pyrogram.api.functions.messages.**GetFeaturedStickers**

**ID**

0x2dacca4f

**Parameters** **hash** – int 32-bit

**Returns** Either *messages.FeaturedStickersNotModified* or *messages.FeaturedStickers*

### GetFullChat

**class** pyrogram.api.functions.messages.**GetFullChat**

**ID**

0x3b831c66

**Parameters** **chat\_id** – int 32-bit

**Returns** *messages.ChatFull*

### GetGameHighScores

**class** pyrogram.api.functions.messages.**GetGameHighScores**

**ID**

0xe822649d

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **id** – int 32-bit
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** *messages.HighScores*

## GetHistory

**class** pyrogram.api.functions.messages.**GetHistory**

**ID**

0xdcbb8260

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **offset\_id** – int 32-bit
- **offset\_date** – int 32-bit
- **add\_offset** – int 32-bit
- **limit** – int 32-bit
- **max\_id** – int 32-bit
- **min\_id** – int 32-bit
- **hash** – int 32-bit

**Returns** Either *messages.Messages*, *messages.MessagesSlice*, *messages.ChannelMessages* or *messages.MessagesNotModified*

## GetInlineBotResults

**class** pyrogram.api.functions.messages.**GetInlineBotResults**

**ID**

0x514e999d

### Parameters

- **bot** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **query** – str
- **offset** – str
- **geo\_point** (*optional*) – Either *InputGeoPointEmpty* or *InputGeoPoint*

**Returns** *messages.BotResults*

## GetInlineGameHighScores

**class** pyrogram.api.functions.messages.**GetInlineGameHighScores**

**ID**

0x0f635e1b

**Parameters**

- **id** – *InputBotInlineMessageID*
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** *messages.HighScores*

**GetMaskStickers**

**class** `pyrogram.api.functions.messages.GetMaskStickers`

**ID**

0x65b8c79f

**Parameters** **hash** – int 32-bit

**Returns** Either *messages.AllStickersNotModified* or *messages.AllStickers*

**GetMessageEditData**

**class** `pyrogram.api.functions.messages.GetMessageEditData`

**ID**

0xfda68d36

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **id** – int 32-bit

**Returns** *messages.MessageEditData*

**GetMessages**

**class** `pyrogram.api.functions.messages.GetMessages`

**ID**

0x63c66506

**Parameters** **id** – List of either *InputMessageID*, *InputMessageReplyTo* or *InputMessagePinned*

**Returns** Either *messages.Messages*, *messages.MessagesSlice*, *messages.ChannelMessages* or *messages.MessagesNotModified*

**GetMessagesViews**

**class** `pyrogram.api.functions.messages.GetMessagesViews`

**ID**

0xc4c8a55d

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **id** – List of int 32-bit
- **increment** – bool

**Returns** List of int 32-bit

## GetOnlines

**class** pyrogram.api.functions.messages.**GetOnlines**

**ID**

0x6e2be050

**Parameters** **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*

**Returns** *ChatOnlines*

## GetPeerDialogs

**class** pyrogram.api.functions.messages.**GetPeerDialogs**

**ID**

0xe470bcfd

**Parameters** **peers** – List of *InputDialogPeer*

**Returns** *messages.PeerDialogs*

## GetPeerSettings

**class** pyrogram.api.functions.messages.**GetPeerSettings**

**ID**

0x3672e09c

**Parameters** **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*

**Returns** *PeerSettings*



### GetPinnedDialogs

**class** pyrogram.api.functions.messages.**GetPinnedDialogs**

**ID**

0xe254d64e

No parameters required.

**Returns** *messages.PeerDialogs*

### GetPollResults

**class** pyrogram.api.functions.messages.**GetPollResults**

**ID**

0x73bb643b

#### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **msg\_id** – int 32-bit

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

### GetRecentLocations

**class** pyrogram.api.functions.messages.**GetRecentLocations**

**ID**

0xbbc45b09

#### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **limit** – int 32-bit
- **hash** – int 32-bit

**Returns** Either *messages.Messages*, *messages.MessagesSlice*, *messages.ChannelMessages* or *messages.MessagesNotModified*

### GetRecentStickers

**class** pyrogram.api.functions.messages.**GetRecentStickers**

**ID**

0x5ea192c9

**Parameters**

- **hash** – int 32-bit
- **attached** (*optional*) – bool

**Returns** Either *messages.RecentStickersNotModified* or *messages.RecentStickers*

**GetSavedGifs**

```
class pyrogram.api.functions.messages.GetSavedGifs
```

**ID**

0x83bf3d52

**Parameters** **hash** – int 32-bit

**Returns** Either *messages.SavedGifsNotModified* or *messages.SavedGifs*

**GetSplitRanges**

```
class pyrogram.api.functions.messages.GetSplitRanges
```

**ID**

0x1cff7e08

No parameters required.

**Returns** List of *MessageRange*

**GetStatsURL**

```
class pyrogram.api.functions.messages.GetStatsURL
```

**ID**

0x812c2ae6

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **params** – str
- **dark** (*optional*) – bool

**Returns** *StatsURL*

**GetStickerSet**

```
class pyrogram.api.functions.messages.GetStickerSet
```

**ID**

0x2619a90e

**Parameters** **stickerset** – Either *InputStickerSetEmpty*, *InputStickerSetID* or *InputStickerSetShortName*

**Returns** *messages.StickerSet*

## GetStickers

**class** pyrogram.api.functions.messages.**GetStickers**

**ID**

0x043d4f2c

**Parameters**

- **emoticon** – str
- **hash** – int 32-bit

**Returns** Either *messages.StickersNotModified* or *messages.Stickers*

## GetUnreadMentions

**class** pyrogram.api.functions.messages.**GetUnreadMentions**

**ID**

0x46578472

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **offset\_id** – int 32-bit
- **add\_offset** – int 32-bit
- **limit** – int 32-bit
- **max\_id** – int 32-bit
- **min\_id** – int 32-bit

**Returns** Either *messages.Messages*, *messages.MessagesSlice*, *messages.ChannelMessages* or *messages.MessagesNotModified*

## GetWebPage

**class** pyrogram.api.functions.messages.**GetWebPage**

**ID**

0x32ca8f91

**Parameters**

- **url** – str
- **hash** – int 32-bit

**Returns** Either *WebPageEmpty*, *WebPagePending*, *WebPage* or *WebPageNotModified*

## GetWebPagePreview

**class** pyrogram.api.functions.messages.**GetWebPagePreview**

**ID**

0x8b68b0cc

**Parameters**

- **message** – str
- **entities** (optional) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

**Returns** Either *MessageMediaEmpty*, *MessageMediaPhoto*, *MessageMediaGeo*, *MessageMediaContact*, *MessageMediaUnsupported*, *MessageMediaDocument*, *MessageMediaWebPage*, *MessageMediaVenue*, *MessageMediaGame*, *MessageMediaInvoice*, *MessageMediaGeoLive* or *MessageMediaPoll*

## HideReportSpam

**class** pyrogram.api.functions.messages.**HideReportSpam**

**ID**

0xa8f1709b

**Parameters** **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*

**Returns** bool

## ImportChatInvite

**class** pyrogram.api.functions.messages.**ImportChatInvite**

**ID**

0x6c50051c

**Parameters** **hash** – str

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

### InstallStickerSet

**class** pyrogram.api.functions.messages.**InstallStickerSet**

**ID**

0xc78fe460

**Parameters**

- **stickerset** – Either *InputStickerSetEmpty*, *InputStickerSetID* or *InputStickerSetShortName*
- **archived** – bool

**Returns** Either *messages.StickerSetInstallResultSuccess* or *messages.StickerSetInstallResultArchive*

### MarkDialogUnread

**class** pyrogram.api.functions.messages.**MarkDialogUnread**

**ID**

0xc286d98f

**Parameters**

- **peer** – *InputDialogPeer*
- **unread**(*optional*) – bool

**Returns** bool

### MigrateChat

**class** pyrogram.api.functions.messages.**MigrateChat**

**ID**

0x15a3b8e3

**Parameters** **chat\_id** – int 32-bit

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

### ReadEncryptedHistory

```
class pyrogram.api.functions.messages.ReadEncryptedHistory
```

**ID**  
0x7f4b690a

**Parameters**

- **peer** – *InputEncryptedChat*
- **max\_date** – int 32-bit

**Returns** bool

### ReadFeaturedStickers

```
class pyrogram.api.functions.messages.ReadFeaturedStickers
```

**ID**  
0x5b118126

**Parameters** **id** – List of int 64-bit

**Returns** bool

### ReadHistory

```
class pyrogram.api.functions.messages.ReadHistory
```

**ID**  
0x0e306d3a

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **max\_id** – int 32-bit

**Returns** *messages.AffectedMessages*

### ReadMentions

```
class pyrogram.api.functions.messages.ReadMentions
```

**ID**  
0x0f0189d3

**Parameters** **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*

**Returns** *messages.AffectedHistory*

### ReadMessageContents

**class** pyrogram.api.functions.messages.**ReadMessageContents**

**ID**

0x36a73f77

**Parameters** **id** – List of int 32-bit

**Returns** *messages.AffectedMessages*

### ReceivedMessages

**class** pyrogram.api.functions.messages.**ReceivedMessages**

**ID**

0x05a954c0

**Parameters** **max\_id** – int 32-bit

**Returns** List of *ReceivedNotifyMessage*

### ReceivedQueue

**class** pyrogram.api.functions.messages.**ReceivedQueue**

**ID**

0x55a5bb66

**Parameters** **max\_qts** – int 32-bit

**Returns** List of int 64-bit

### ReorderPinnedDialogs

**class** pyrogram.api.functions.messages.**ReorderPinnedDialogs**

**ID**

0x5b51d63f

**Parameters**

- **order** – List of *InputDialogPeer*
- **force** (*optional*) – bool

**Returns** bool

## ReorderStickerSets

**class** pyrogram.api.functions.messages.**ReorderStickerSets**

**ID**

0x78337739

### Parameters

- **order** – List of int 64-bit
- **masks** (*optional*) – bool

**Returns** bool

## Report

**class** pyrogram.api.functions.messages.**Report**

**ID**

0xbd82b658

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **id** – List of int 32-bit
- **reason** – Either *InputReportReasonSpam*, *InputReportReasonViolence*, *InputReportReasonPornography*, *InputReportReasonChildAbuse*, *InputReportReasonOther* or *InputReportReasonCopyright*

**Returns** bool

## ReportEncryptedSpam

**class** pyrogram.api.functions.messages.**ReportEncryptedSpam**

**ID**

0x4b0c8c0f

**Parameters** **peer** – *InputEncryptedChat*

**Returns** bool

## ReportSpam

**class** pyrogram.api.functions.messages.**ReportSpam**

**ID**

0xcf1592db



**Parameters** **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*

**Returns** bool

## RequestEncryption

**class** pyrogram.api.functions.messages.**RequestEncryption**

**ID**

0xf64daf43

**Parameters**

- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **random\_id** – int 32-bit
- **g\_a** – bytes

**Returns** Either *EncryptedChatEmpty*, *EncryptedChatWaiting*, *EncryptedChatRequested*, *EncryptedChat* or *EncryptedChatDiscarded*

## SaveDraft

**class** pyrogram.api.functions.messages.**SaveDraft**

**ID**

0xbc39e14b

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **message** – str
- **no\_webpage** (*optional*) – bool
- **reply\_to\_msg\_id** (*optional*) – int 32-bit
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

**Returns** bool

## SaveGif

**class** pyrogram.api.functions.messages.**SaveGif**

**ID**

0x327a30cb

**Parameters**

- **id** – Either *InputDocumentEmpty* or *InputDocument*
- **unsave** – bool

**Returns** bool**SaveRecentSticker****class** pyrogram.api.functions.messages.**SaveRecentSticker****ID**

0x392718f8

**Parameters**

- **id** – Either *InputDocumentEmpty* or *InputDocument*
- **unsave** – bool
- **attached** (*optional*) – bool

**Returns** bool**Search****class** pyrogram.api.functions.messages.**Search****ID**

0x8614ef68

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **q** – str
- **filter** – Either *InputMessagesFilterEmpty*, *InputMessagesFilterPhotos*, *InputMessagesFilterPhotoVideo*, *InputMessagesFilterDocument*, *InputMessagesFilterUrl*, *InputMessagesFilterGif*, *InputMessagesFilterVoice*, *InputMessagesFilterMusic*, *InputMessagesFilterChatPhotos*, *InputMessagesFilterPhoneCalls*, *InputMessagesFilterRoundVoice*, *InputMessagesFilterRoundVideo*, *InputMessagesFilterMyMentions*, *InputMessagesFilterGeo* or *InputMessagesFilterContacts*
- **min\_date** – int 32-bit
- **max\_date** – int 32-bit
- **offset\_id** – int 32-bit
- **add\_offset** – int 32-bit

- **limit** – int 32-bit
- **max\_id** – int 32-bit
- **min\_id** – int 32-bit
- **hash** – int 32-bit
- **from\_id** (optional) – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** Either *messages.Messages*, *messages.MessagesSlice*, *messages.ChannelMessages* or *messages.MessagesNotModified*

## SearchGifs

**class** pyrogram.api.functions.messages.**SearchGifs**

**ID**

0xbf9a776b

### Parameters

- **q** – str
- **offset** – int 32-bit

**Returns** *messages.FoundGifs*

## SearchGlobal

**class** pyrogram.api.functions.messages.**SearchGlobal**

**ID**

0x9e3cacb0

### Parameters

- **q** – str
- **offset\_date** – int 32-bit
- **offset\_peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **offset\_id** – int 32-bit
- **limit** – int 32-bit

**Returns** Either *messages.Messages*, *messages.MessagesSlice*, *messages.ChannelMessages* or *messages.MessagesNotModified*

## SearchStickerSets

**class** pyrogram.api.functions.messages.**SearchStickerSets**

**ID**`0xc2b7d08b`**Parameters**

- **q** – `str`
- **hash** – `int` 32-bit
- **exclude\_featured** (*optional*) – `bool`

**Returns** Either `messages.FoundStickerSetsNotModified` or `messages.FoundStickerSets`

**SendEncrypted**

```
class pyrogram.api.functions.messages.SendEncrypted
```

**ID**`0xa9776773`**Parameters**

- **peer** – `InputEncryptedChat`
- **random\_id** – `int` 64-bit
- **data** – `bytes`

**Returns** Either `messages.SentEncryptedMessage` or `messages.SentEncryptedFile`

**SendEncryptedFile**

```
class pyrogram.api.functions.messages.SendEncryptedFile
```

**ID**`0x9a901b66`**Parameters**

- **peer** – `InputEncryptedChat`
- **random\_id** – `int` 64-bit
- **data** – `bytes`
- **file** – Either `InputEncryptedFileEmpty`, `InputEncryptedFileUploaded`, `InputEncryptedFile` or `InputEncryptedFileBigUploaded`

**Returns** Either `messages.SentEncryptedMessage` or `messages.SentEncryptedFile`

## SendEncryptedService

```
class pyrogram.api.functions.messages.SendEncryptedService
```

**ID**

0x32d439a4

### Parameters

- **peer** – *InputEncryptedChat*
- **random\_id** – int 64-bit
- **data** – bytes

**Returns** Either *messages.SentEncryptedMessage* or *messages.SentEncryptedFile*

## SendInlineBotResult

```
class pyrogram.api.functions.messages.SendInlineBotResult
```

**ID**

0xb16e06fe

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **random\_id** – int 64-bit
- **query\_id** – int 64-bit
- **id** – str
- **silent** (*optional*) – bool
- **background** (*optional*) – bool
- **clear\_draft** (*optional*) – bool
- **hide\_via** (*optional*) – bool
- **reply\_to\_msg\_id** (*optional*) – int 32-bit

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## SendMedia

```
class pyrogram.api.functions.messages.SendMedia
```

**ID**

0xb8d1262b

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **media** – Either *InputMediaEmpty*, *InputMediaUploadedPhoto*, *InputMediaPhoto*, *InputMediaGeoPoint*, *InputMediaContact*, *InputMediaUploadedDocument*, *InputMediaDocument*, *InputMediaVenue*, *InputMediaGifExternal*, *InputMediaPhotoExternal*, *InputMediaDocumentExternal*, *InputMediaGame*, *InputMediaInvoice*, *InputMediaGeoLive* or *InputMediaPoll*
- **message** – str
- **random\_id** – int 64-bit
- **silent** (*optional*) – bool
- **background** (*optional*) – bool
- **clear\_draft** (*optional*) – bool
- **reply\_to\_msg\_id** (*optional*) – int 32-bit
- **reply\_markup** (*optional*) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSendMessage*

## SendMessage

**class** pyrogram.api.functions.messages.**SendMessage**

**ID**

0xfa88427a

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **message** – str
- **random\_id** – int 64-bit
- **no\_webpage** (*optional*) – bool
- **silent** (*optional*) – bool
- **background** (*optional*) – bool
- **clear\_draft** (*optional*) – bool

- **reply\_to\_msg\_id**(*optional*) – int 32-bit
- **reply\_markup** (*optional*) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## SendMultiMedia

**class** pyrogram.api.functions.messages.**SendMultiMedia**

**ID**

0x2095512f

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **multi\_media** – List of *InputSingleMedia*
- **silent**(*optional*) – bool
- **background**(*optional*) – bool
- **clear\_draft**(*optional*) – bool
- **reply\_to\_msg\_id**(*optional*) – int 32-bit

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## SendScreenshotNotification

**class** pyrogram.api.functions.messages.**SendScreenshotNotification**

**ID**

0xc97df020

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **reply\_to\_msg\_id** – int 32-bit

- **random\_id** – int 64-bit

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## SendVote

**class** pyrogram.api.functions.messages.**SendVote**

**ID**

0x10ea6184

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*,  
*InputPeerUser* or *InputPeerChannel*
- **msg\_id** – int 32-bit
- **options** – List of bytes

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*,  
*UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or  
*UpdateShortSentMessage*

## SetBotCallbackAnswer

**class** pyrogram.api.functions.messages.**SetBotCallbackAnswer**

**ID**

0xd58f130a

### Parameters

- **query\_id** – int 64-bit
- **cache\_time** – int 32-bit
- **alert** (*optional*) – bool
- **message** (*optional*) – str
- **url** (*optional*) – str

**Returns** bool

## SetBotPrecheckoutResults

**class** pyrogram.api.functions.messages.**SetBotPrecheckoutResults**

**ID**

0x09c2dd95

### Parameters



- **query\_id** – int 64-bit
- **success** (*optional*) – bool
- **error** (*optional*) – str

**Returns** bool

### SetBotShippingResults

**class** pyrogram.api.functions.messages.SetBotShippingResults

**ID**

0xe5f672fa

#### Parameters

- **query\_id** – int 64-bit
- **error** (*optional*) – str
- **shipping\_options** (*optional*) – List of *ShippingOption*

**Returns** bool

### SetEncryptedTyping

**class** pyrogram.api.functions.messages.SetEncryptedTyping

**ID**

0x791451ed

#### Parameters

- **peer** – *InputEncryptedChat*
- **typing** – bool

**Returns** bool

### SetGameScore

**class** pyrogram.api.functions.messages.SetGameScore

**ID**

0x8ef8ecc0

#### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **id** – int 32-bit
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **score** – int 32-bit

- **edit\_message**(*optional*) – bool
- **force**(*optional*) – bool

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSendMessage*

## SetInlineBotResults

**class** pyrogram.api.functions.messages.**SetInlineBotResults**

**ID**

0xeb5ea206

### Parameters

- **query\_id** – int 64-bit
- **results** – List of either *InputBotInlineResult*, *InputBotInlineResultPhoto*, *InputBotInlineResultDocument* or *InputBotInlineResultGame*
- **cache\_time** – int 32-bit
- **gallery**(*optional*) – bool
- **private**(*optional*) – bool
- **next\_offset**(*optional*) – str
- **switch\_pm**(*optional*) – *InlineBotSwitchPM*

**Returns** bool

## SetInlineGameScore

**class** pyrogram.api.functions.messages.**SetInlineGameScore**

**ID**

0x15ad9f64

### Parameters

- **id** – *InputBotInlineMessageID*
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **score** – int 32-bit
- **edit\_message**(*optional*) – bool
- **force**(*optional*) – bool

**Returns** bool

## SetTyping

**class** pyrogram.api.functions.messages.**SetTyping**

**ID**

0xa3825e50

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **action** – Either *SendMessageTypingAction*, *SendMessageCancelAction*, *SendMessageRecordVideoAction*, *SendMessageUploadVideoAction*, *SendMessageRecordAudioAction*, *SendMessageUploadAudioAction*, *SendMessageUploadPhotoAction*, *SendMessageUploadDocumentAction*, *SendMessageGeoLocationAction*, *SendMessageChooseContactAction*, *SendMessageGamePlayAction*, *SendMessageRecordRoundAction* or *SendMessageUploadRoundAction*

**Returns** bool

## StartBot

**class** pyrogram.api.functions.messages.**StartBot**

**ID**

0xe6df7378

### Parameters

- **bot** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **random\_id** – int 64-bit
- **start\_param** – str

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## ToggleDialogPin

**class** pyrogram.api.functions.messages.**ToggleDialogPin**

**ID**

0xa731e257

### Parameters

- **peer** – *InputDialogPeer*
- **pinned** (*optional*) – bool

**Returns** bool

### UninstallStickerSet

**class** pyrogram.api.functions.messages.UninstallStickerSet

**ID**

0xf96e55de

**Parameters** **stickerset** – Either *InputStickerSetEmpty*, *InputStickerSetID* or *InputStickerSetShortName*

**Returns** bool

### UpdatePinnedMessage

**class** pyrogram.api.functions.messages.UpdatePinnedMessage

**ID**

0xd2aaf7ec

**Parameters**

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **id** – int 32-bit
- **silent** (optional) – bool

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

### UploadEncryptedFile

**class** pyrogram.api.functions.messages.UploadEncryptedFile

**ID**

0x5057c497

**Parameters**

- **peer** – *InputEncryptedChat*
- **file** – Either *InputEncryptedFileEmpty*, *InputEncryptedFileUploaded*, *InputEncryptedFile* or *InputEncryptedFileBigUploaded*

**Returns** Either *EncryptedFileEmpty* or *EncryptedFile*

## UploadMedia

**class** pyrogram.api.functions.messages.UploadMedia

**ID**

0x519bc2b1

### Parameters

- **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*
- **media** – Either *InputMediaEmpty*, *InputMediaUploadedPhoto*, *InputMediaPhoto*, *InputMediaGeoPoint*, *InputMediaContact*, *InputMediaUploadedDocument*, *InputMediaDocument*, *InputMediaVenue*, *InputMediaGifExternal*, *InputMediaPhotoExternal*, *InputMediaDocumentExternal*, *InputMediaGame*, *InputMediaInvoice*, *InputMediaGeoLive* or *InputMediaPoll*

**Returns** Either *MessageMediaEmpty*, *MessageMediaPhoto*, *MessageMediaGeo*, *MessageMediaContact*, *MessageMediaUnsupported*, *MessageMediaDocument*, *MessageMediaWebPage*, *MessageMediaVenue*, *MessageMediaGame*, *MessageMediaInvoice*, *MessageMediaGeoLive* or *MessageMediaPoll*

## 4.36.10 Payments

### ClearSavedInfo

**class** pyrogram.api.functions.payments.ClearSavedInfo

**ID**

0xd83d70c1

### Parameters

- **credentials** (*optional*) – bool
- **info** (*optional*) – bool

**Returns** bool

### GetPaymentForm

**class** pyrogram.api.functions.payments.GetPaymentForm

**ID**

0x99f09745

**Parameters** *msg\_id* – int 32-bit

**Returns** *payments.PaymentForm*

### GetPaymentReceipt

**class** pyrogram.api.functions.payments.**GetPaymentReceipt**

**ID**

0xa092a980

**Parameters** *msg\_id* – int 32-bit

**Returns** *payments.PaymentReceipt*

### GetSavedInfo

**class** pyrogram.api.functions.payments.**GetSavedInfo**

**ID**

0x227d824b

No parameters required.

**Returns** *payments.SavedInfo*

### SendPaymentForm

**class** pyrogram.api.functions.payments.**SendPaymentForm**

**ID**

0x2b8879b3

**Parameters**

- *msg\_id* – int 32-bit
- **credentials** – Either *InputPaymentCredentialsSaved*, *InputPaymentCredentials*, *InputPaymentCredentialsApplePay* or *InputPaymentCredentialsAndroidPay*
- *requested\_info\_id*(*optional*) – str
- *shipping\_option\_id*(*optional*) – str

**Returns** Either *payments.PaymentResult* or *payments.PaymentVerificationNeeded*

### ValidateRequestedInfo

**class** pyrogram.api.functions.payments.**ValidateRequestedInfo**

**ID**

0x770a8e74

**Parameters**

- *msg\_id* – int 32-bit

- **info** – *PaymentRequestedInfo*
- **save** (*optional*) – bool

Returns *payments.ValidatedRequestedInfo*

### 4.36.11 Phone

#### AcceptCall

**class** pyrogram.api.functions.phone.**AcceptCall**

**ID**

0x3bd2b4a0

#### Parameters

- **peer** – *InputPhoneCall*
- **g\_b** – bytes
- **protocol** – *PhoneCallProtocol*

Returns *phone.PhoneCall*

#### ConfirmCall

**class** pyrogram.api.functions.phone.**ConfirmCall**

**ID**

0x2efe1722

#### Parameters

- **peer** – *InputPhoneCall*
- **g\_a** – bytes
- **key\_fingerprint** – int 64-bit
- **protocol** – *PhoneCallProtocol*

Returns *phone.PhoneCall*

#### DiscardCall

**class** pyrogram.api.functions.phone.**DiscardCall**

**ID**

0x78d413a6

#### Parameters

- **peer** – *InputPhoneCall*
- **duration** – int 32-bit

- **reason** – Either *PhoneCallDiscardReasonMissed*, *PhoneCallDiscardReasonDisconnect*, *PhoneCallDiscardReasonHangup* or *PhoneCallDiscardReasonBusy*
- **connection\_id** – int 64-bit

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## GetCallConfig

**class** pyrogram.api.functions.phone.**GetCallConfig**

**ID**  
0x55451fa9

No parameters required.

**Returns** *DataJSON*

## ReceivedCall

**class** pyrogram.api.functions.phone.**ReceivedCall**

**ID**  
0x17d54f61

**Parameters** **peer** – *InputPhoneCall*

**Returns** bool

## RequestCall

**class** pyrogram.api.functions.phone.**RequestCall**

**ID**  
0x5b95b3d4

**Parameters**

- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **random\_id** – int 32-bit
- **g\_a\_hash** – bytes
- **protocol** – *PhoneCallProtocol*

**Returns** *phone.PhoneCall*



### SaveCallDebug

```
class pyrogram.api.functions.phone.SaveCallDebug
```

**ID**

0x277add7e

**Parameters**

- **peer** – *InputPhoneCall*
- **debug** – *DataJSON*

**Returns** bool

### SetCallRating

```
class pyrogram.api.functions.phone.SetCallRating
```

**ID**

0x59ead627

**Parameters**

- **peer** – *InputPhoneCall*
- **rating** – int 32-bit
- **comment** – str
- **user\_initiative** (*optional*) – bool

**Returns** Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

## 4.36.12 Photos

### DeletePhotos

```
class pyrogram.api.functions.photos.DeletePhotos
```

**ID**

0x87cf7f2f

**Parameters** **id** – List of either *InputPhotoEmpty* or *InputPhoto*

**Returns** List of int 64-bit

### GetUserPhotos

```
class pyrogram.api.functions.photos.GetUserPhotos
```

**ID**

0x91cd32a8

**Parameters**

- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **offset** – int 32-bit
- **max\_id** – int 64-bit
- **limit** – int 32-bit

**Returns** Either *photos.Photos* or *photos.PhotosSlice***UpdateProfilePhoto****class** pyrogram.api.functions.photos.**UpdateProfilePhoto****ID**

0xf0bb5152

**Parameters** **id** – Either *InputPhotoEmpty* or *InputPhoto***Returns** Either *UserProfilePhotoEmpty* or *UserProfilePhoto***UploadProfilePhoto****class** pyrogram.api.functions.photos.**UploadProfilePhoto****ID**

0x4f32c098

**Parameters** **file** – Either *InputFile* or *InputFileBig***Returns** *photos.Photo***4.36.13 Stickers****AddStickerToSet****class** pyrogram.api.functions.stickers.**AddStickerToSet****ID**

0x8653febe

**Parameters**

- **stickerset** – Either *InputStickerSetEmpty*, *InputStickerSetID* or *InputStickerSetShortName*
- **sticker** – *InputStickerSetItem*

**Returns** *messages.StickerSet*

### ChangeStickerPosition

**class** pyrogram.api.functions.stickers.ChangeStickerPosition

**ID**

0xffb6d4ca

**Parameters**

- **sticker** – Either *InputDocumentEmpty* or *InputDocument*
- **position** – int 32-bit

**Returns** *messages.StickerSet*

### CreateStickerSet

**class** pyrogram.api.functions.stickers.CreateStickerSet

**ID**

0x9bd86e6a

**Parameters**

- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **title** – str
- **short\_name** – str
- **stickers** – List of *InputStickerSetItem*
- **masks** (*optional*) – bool

**Returns** *messages.StickerSet*

### RemoveStickerFromSet

**class** pyrogram.api.functions.stickers.RemoveStickerFromSet

**ID**

0xf7760f51

**Parameters** **sticker** – Either *InputDocumentEmpty* or *InputDocument*

**Returns** *messages.StickerSet*

## 4.36.14 Updates

### GetChannelDifference

**class** pyrogram.api.functions.updates.GetChannelDifference

**ID**

0x03173d78

**Parameters**

- **channel** – Either *InputChannelEmpty* or *InputChannel*
- **filter** – Either *ChannelMessagesFilterEmpty* or *ChannelMessagesFilter*
- **pts** – int 32-bit
- **limit** – int 32-bit
- **force** (*optional*) – bool

**Returns** Either *updates.ChannelDifferenceEmpty*, *updates.ChannelDifferenceTooLong* or *updates.ChannelDifference*

**GetDifference**

```
class pyrogram.api.functions.updates.GetDifference
```

**ID**

0x25939651

**Parameters**

- **pts** – int 32-bit
- **date** – int 32-bit
- **qts** – int 32-bit
- **pts\_total\_limit** (*optional*) – int 32-bit

**Returns** Either *updates.DifferenceEmpty*, *updates.Difference*, *updates.DifferenceSlice* or *updates.DifferenceTooLong*

**GetState**

```
class pyrogram.api.functions.updates.GetState
```

**ID**

0xedd4882a

No parameters required.

**Returns** *updates.State*

**4.36.15 Upload****GetCdnFile**

```
class pyrogram.api.functions.upload.GetCdnFile
```

**ID**

0x2000bcc3

**Parameters**

- **file\_token** – bytes
- **offset** – int 32-bit
- **limit** – int 32-bit

**Returns** Either *upload.CdnFileReuploadNeeded* or *upload.CdnFile***GetCdnFileHashes****class** pyrogram.api.functions.upload.**GetCdnFileHashes****ID**

0x4da54231

**Parameters**

- **file\_token** – bytes
- **offset** – int 32-bit

**Returns** List of *FileHash***GetFile****class** pyrogram.api.functions.upload.**GetFile****ID**

0xe3a6cfb5

**Parameters**

- **location** – Either *InputFileLocation*, *InputEncryptedFileLocation*, *InputDocumentFileLocation*, *InputSecureFileLocation* or *InputTakeoutFileLocation*
- **offset** – int 32-bit
- **limit** – int 32-bit

**Returns** Either *upload.File* or *upload.FileCdnRedirect***GetFileHashes****class** pyrogram.api.functions.upload.**GetFileHashes****ID**

0xc7025931

**Parameters**

- **location** – Either *InputFileLocation*, *InputEncryptedFileLocation*, *InputDocumentFileLocation*, *InputSecureFileLocation* or *InputTakeoutFileLocation*
- **offset** – int 32-bit

**Returns** List of *FileHash*

### GetWebFile

**class** pyrogram.api.functions.upload.**GetWebFile**

**ID**

0x24e6818d

**Parameters**

- **location** – Either *InputWebFileLocation* or *InputWebFileGeoPointLocation*
- **offset** – int 32-bit
- **limit** – int 32-bit

**Returns** *upload.WebFile*

### ReuploadCdnFile

**class** pyrogram.api.functions.upload.**ReuploadCdnFile**

**ID**

0x9b2754a8

**Parameters**

- **file\_token** – bytes
- **request\_token** – bytes

**Returns** List of *FileHash*

### SaveBigFilePart

**class** pyrogram.api.functions.upload.**SaveBigFilePart**

**ID**

0xde7b673d

**Parameters**

- **file\_id** – int 64-bit
- **file\_part** – int 32-bit
- **file\_total\_parts** – int 32-bit
- **bytes** – bytes

**Returns** bool

### SaveFilePart

**class** pyrogram.api.functions.upload.**SaveFilePart**

**ID**

0xb304a621

**Parameters**

- **file\_id** – int 64-bit
- **file\_part** – int 32-bit
- **bytes** – bytes

**Returns** bool

## 4.36.16 Users

### GetFullUser

**class** pyrogram.api.functions.users.**GetFullUser**

**ID**

0xca30a5b1

**Parameters** **id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** *UserFull*

### GetUsers

**class** pyrogram.api.functions.users.**GetUsers**

**ID**

0x0d91a548

**Parameters** **id** – List of either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**Returns** List of either *UserEmpty* or *User*

### SetSecureValueErrors

**class** pyrogram.api.functions.users.**SetSecureValueErrors**

**ID**

0x90c894b5

**Parameters**

- **id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **errors** – List of either *SecureValueErrorData*, *SecureValueErrorFrontSide*, *SecureValueErrorReverseSide*, *SecureValueErrorSelfie*, *SecureValueErrorFile*, *SecureValueErrorFiles*, *SecureValueError*, *SecureValueErrorTranslationFile* or *SecureValueErrorTranslationFiles*

**Returns** bool

#### 4.36.17 DestroyAuthKey

**class** pyrogram.api.functions.DestroyAuthKey

**ID**

0xd1435160

No parameters required.

**Returns** Either *DestroyAuthKeyOk*, *DestroyAuthKeyNone* or *DestroyAuthKeyFail*

#### 4.36.18 DestroySession

**class** pyrogram.api.functions.DestroySession

**ID**

0xe7512126

**Parameters** *session\_id* – int 64-bit

**Returns** Either *DestroySessionOk* or *DestroySessionNone*

#### 4.36.19 GetFutureSalts

**class** pyrogram.api.functions.GetFutureSalts

**ID**

0xb921bd04

**Parameters** *num* – int 32-bit

**Returns** FutureSalts

#### 4.36.20 InitConnection

**class** pyrogram.api.functions.InitConnection

**ID**

0x785188b8

**Parameters**



- **api\_id** – int 32-bit
- **device\_model** – str
- **system\_version** – str
- **app\_version** – str
- **system\_lang\_code** – str
- **lang\_pack** – str
- **lang\_code** – str
- **query** – Any method from *pyrogram.api.functions*
- **proxy** (*optional*) – *InputClientProxy*

**Returns** Any object from *pyrogram.api.types*

#### 4.36.21 InvokeAfterMsg

**class** `pyrogram.api.functions.InvokeAfterMsg`

**ID**

0xcb9f372d

**Parameters**

- **msg\_id** – int 64-bit
- **query** – Any method from *pyrogram.api.functions*

**Returns** Any object from *pyrogram.api.types*

#### 4.36.22 InvokeAfterMsgs

**class** `pyrogram.api.functions.InvokeAfterMsgs`

**ID**

0x3dc4b4f0

**Parameters**

- **msg\_ids** – List of int 64-bit
- **query** – Any method from *pyrogram.api.functions*

**Returns** Any object from *pyrogram.api.types*

#### 4.36.23 InvokeWithLayer

**class** `pyrogram.api.functions.InvokeWithLayer`

**ID**

0xda9b0d0d

**Parameters**

- **layer** – int 32-bit
- **query** – Any method from *pyrogram.api.functions*

**Returns** Any object from *pyrogram.api.types*

#### 4.36.24 InvokeWithMessagesRange

**class** `pyrogram.api.functions.InvokeWithMessagesRange`

**ID**

0x365275f2

**Parameters**

- **range** – *MessageRange*
- **query** – Any method from *pyrogram.api.functions*

**Returns** Any object from *pyrogram.api.types*

#### 4.36.25 InvokeWithTakeout

**class** `pyrogram.api.functions.InvokeWithTakeout`

**ID**

0xaca9fd2e

**Parameters**

- **takeout\_id** – int 64-bit
- **query** – Any method from *pyrogram.api.functions*

**Returns** Any object from *pyrogram.api.types*

#### 4.36.26 InvokeWithoutUpdates

**class** `pyrogram.api.functions.InvokeWithoutUpdates`

**ID**

0xbf9459b7

**Parameters** **query** – Any method from *pyrogram.api.functions*

**Returns** Any object from *pyrogram.api.types*

### 4.36.27 Ping

**class** pyrogram.api.functions.Ping

**ID**

0x7abe77ec

**Parameters** `ping_id` – int 64-bit

**Returns** *Pong*

### 4.36.28 PingDelayDisconnect

**class** pyrogram.api.functions.PingDelayDisconnect

**ID**

0xf3427b8c

**Parameters**

- `ping_id` – int 64-bit
- `disconnect_delay` – int 32-bit

**Returns** *Pong*

### 4.36.29 ReqDHParams

**class** pyrogram.api.functions.ReqDHParams

**ID**

0xd712e4be

**Parameters**

- `nonce` – int 128-bit
- `server_nonce` – int 128-bit
- `p` – bytes
- `q` – bytes
- `public_key_fingerprint` – int 64-bit
- `encrypted_data` – bytes

**Returns** Either *ServerDHParamsFail* or *ServerDHParamsOk*

### 4.36.30 ReqPq

**class** pyrogram.api.functions.ReqPq

**ID**

0x60469778

**Parameters** `nonce` – int 128-bit

**Returns** *ResPQ*

### 4.36.31 ReqPqMulti

**class** `pyrogram.api.functions.ReqPqMulti`

**ID**

0xbe7e8ef1

**Parameters** `nonce` – int 128-bit

**Returns** *ResPQ*

### 4.36.32 RpcDropAnswer

**class** `pyrogram.api.functions.RpcDropAnswer`

**ID**

0x58e4a740

**Parameters** `req_msg_id` – int 64-bit

**Returns** Either *RpcAnswerUnknown*, *RpcAnswerDroppedRunning* or *RpcAnswerDropped*

### 4.36.33 SetClientDHParams

**class** `pyrogram.api.functions.SetClientDHParams`

**ID**

0xf5045f1f

**Parameters**

- `nonce` – int 128-bit
- `server_nonce` – int 128-bit
- `encrypted_data` – bytes

**Returns** Either *DhGenOk*, *DhGenRetry* or *DhGenFail*

## 4.37 Types

### 4.37.1 Account

## AuthorizationForm

**class** pyrogram.api.types.account.**AuthorizationForm**

**ID**

0xad2e1cd8

### Parameters

- **required\_types** – List of either *SecureRequiredType* or *SecureRequiredTypeOneOf*
- **values** – List of *SecureValue*
- **errors** – List of either *SecureValueErrorData*, *SecureValueErrorFrontSide*, *SecureValueErrorReverseSide*, *SecureValueErrorSelfie*, *SecureValueErrorFile*, *SecureValueErrorFiles*, *SecureValueError*, *SecureValueErrorTranslationFile* or *SecureValueErrorTranslationFiles*
- **users** – List of either *UserEmpty* or *User*
- **privacy\_policy\_url** (*optional*) – str

**See also:**

This object can be returned by *account.GetAuthorizationForm*.

## Authorizations

**class** pyrogram.api.types.account.**Authorizations**

**ID**

0x1250abde

**Parameters** **authorizations** – List of *Authorization*

**See also:**

This object can be returned by *account.GetAuthorizations*.

## AutoDownloadSettings

**class** pyrogram.api.types.account.**AutoDownloadSettings**

**ID**

0x63cacf26

### Parameters

- **low** – *AutoDownloadSettings*
- **medium** – *AutoDownloadSettings*
- **high** – *AutoDownloadSettings*

**See also:**

This object can be returned by `account.GetAutoDownloadSettings`.

**Password**

**class** pyrogram.api.types.account.Password

**ID**

0xad2641f8

**Parameters**

- **new\_algo** – Either `PasswordKdfAlgoUnknown` or `PasswordKdfAlgoSHA256SHA256PBKDF2HMACSHA512iter100000SHA256ModPow`
- **new\_secure\_algo** – Either `SecurePasswordKdfAlgoUnknown`, `SecurePasswordKdfAlgoPBKDF2HMACSHA512iter100000` or `SecurePasswordKdfAlgoSHA512`
- **secure\_random** – bytes
- **has\_recovery** (*optional*) – bool
- **has\_secure\_values** (*optional*) – bool
- **has\_password** (*optional*) – bool
- **current\_algo** (*optional*) – Either `PasswordKdfAlgoUnknown` or `PasswordKdfAlgoSHA256SHA256PBKDF2HMACSHA512iter100000SHA256ModPow`
- **srp\_B** (*optional*) – bytes
- **srp\_id** (*optional*) – int 64-bit
- **hint** (*optional*) – str
- **email\_unconfirmed\_pattern** (*optional*) – str

**See also:**

This object can be returned by `account.GetPassword`.

**PasswordInputSettings**

**class** pyrogram.api.types.account.PasswordInputSettings

**ID**

0xc23727c9

**Parameters**

- **new\_algo** (*optional*) – Either `PasswordKdfAlgoUnknown` or `PasswordKdfAlgoSHA256SHA256PBKDF2HMACSHA512iter100000SHA256ModPow`
- **new\_password\_hash** (*optional*) – bytes
- **hint** (*optional*) – str
- **email** (*optional*) – str

- **new\_secure\_settings** (*optional*) – *SecureSecretSettings*

## PasswordSettings

**class** pyrogram.api.types.account.**PasswordSettings**

**ID**

0x9a5c33e5

### Parameters

- **email** (*optional*) – *str*
- **secure\_settings** (*optional*) – *SecureSecretSettings*

**See also:**

This object can be returned by *account.GetPasswordSettings*.

## PrivacyRules

**class** pyrogram.api.types.account.**PrivacyRules**

**ID**

0x554abb6f

### Parameters

- **rules** – List of either *PrivacyValueAllowContacts*, *PrivacyValueAllowAll*, *PrivacyValueAllowUsers*, *PrivacyValueDisallowContacts*, *PrivacyValueDisallowAll* or *PrivacyValueDisallowUsers*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *account.GetPrivacy* and *account.SetPrivacy*.

## SentEmailCode

**class** pyrogram.api.types.account.**SentEmailCode**

**ID**

0x811f854f

### Parameters

- **email\_pattern** – *str*
- **length** – *int* 32-bit

**See also:**

This object can be returned by *account.SendVerifyEmailCode*.

## Takeout

**class** pyrogram.api.types.account.**Takeout**

**ID**  
0x4dba4501

**Parameters** **id** – int 64-bit

### See also:

This object can be returned by [\*account.InitTakeoutSession\*](#).

## TmpPassword

**class** pyrogram.api.types.account.**TmpPassword**

**ID**  
0xdb64fd34

### Parameters

- **tmp\_password** – bytes
- **valid\_until** – int 32-bit

### See also:

This object can be returned by [\*account.GetTmpPassword\*](#).

## WallPapers

**class** pyrogram.api.types.account.**WallPapers**

**ID**  
0x702b65a9

### Parameters

- **hash** – int 32-bit
- **wallpapers** – List of [\*WallPaper\*](#)

### See also:

This object can be returned by [\*account.GetWallPapers\*](#).

## WallPapersNotModified

**class** pyrogram.api.types.account.**WallPapersNotModified**

**ID**  
0x1c199183



No parameters required.

**See also:**

This object can be returned by `account.GetWallPapers`.

## WebAuthorizations

**class** pyrogram.api.types.account.WebAuthorizations

**ID**

0xed56c9fc

**Parameters**

- **authorizations** – List of *WebAuthorization*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by `account.GetWebAuthorizations`.

## 4.37.2 Auth

### Authorization

**class** pyrogram.api.types.auth.Authorization

**ID**

0xcd050916

**Parameters**

- **user** – Either *UserEmpty* or *User*
- **tmp\_sessions** (*optional*) – int 32-bit

**See also:**

This object can be returned by `auth.SignUp`, `auth.SignIn`, `auth.ImportAuthorization`, `auth.ImportBotAuthorization`, `auth.CheckPassword` and `auth.RecoverPassword`.

### CheckedPhone

**class** pyrogram.api.types.auth.CheckedPhone

**ID**

0x811ea28e

**Parameters** `phone_registered` – bool

### CodeTypeCall

```
class pyrogram.api.types.auth.CodeTypeCall
```

**ID**

0x741cd3e3

No parameters required.

### CodeTypeFlashCall

```
class pyrogram.api.types.auth.CodeTypeFlashCall
```

**ID**

0x226ccefb

No parameters required.

### CodeTypeSms

```
class pyrogram.api.types.auth.CodeTypeSms
```

**ID**

0x72a3158c

No parameters required.

### ExportedAuthorization

```
class pyrogram.api.types.auth.ExportedAuthorization
```

**ID**

0xdf969c2d

**Parameters**

- **id** – int 32-bit
- **bytes** – bytes

**See also:**

This object can be returned by [\*auth.ExportAuthorization\*](#).

### PasswordRecovery

```
class pyrogram.api.types.auth.PasswordRecovery
```

**ID**

0x137948a5

**Parameters** **email\_pattern** – str

**See also:**

This object can be returned by `auth.RequestPasswordRecovery`.

**SentCode**

```
class pyrogram.api.types.auth.SentCode
```

**ID**

0x38faab5f

**Parameters**

- **type** – Either `auth.SentCodeTypeApp`, `auth.SentCodeTypeSms`, `auth.SentCodeTypeCall` or `auth.SentCodeTypeFlashCall`
- **phone\_code\_hash** – str
- **phone\_registered** (optional) – bool
- **next\_type** (optional) – Either `auth.CodeTypeSms`, `auth.CodeTypeCall` or `auth.CodeTypeFlashCall`
- **timeout** (optional) – int 32-bit
- **terms\_of\_service** (optional) – `help.TermsOfService`

**See also:**

This object can be returned by `auth.SendCode`, `auth.ResendCode`, `account.SendChangePhoneCode`, `account.SendConfirmPhoneCode` and `account.SendVerifyPhoneCode`.

**SentCodeTypeApp**

```
class pyrogram.api.types.auth.SentCodeTypeApp
```

**ID**

0x3dbb5986

**Parameters** **length** – int 32-bit

**SentCodeTypeCall**

```
class pyrogram.api.types.auth.SentCodeTypeCall
```

**ID**

0x5353e5a7

**Parameters** **length** – int 32-bit

### SentCodeTypeFlashCall

```
class pyrogram.api.types.auth.SentCodeTypeFlashCall
```

**ID**

0xab03c6d9

**Parameters** *pattern* – str

### SentCodeTypeSms

```
class pyrogram.api.types.auth.SentCodeTypeSms
```

**ID**

0xc000bba2

**Parameters** *length* – int 32-bit

## 4.37.3 Channels

### AdminLogResults

```
class pyrogram.api.types.channels.AdminLogResults
```

**ID**

0xed8af74d

**Parameters**

- **events** – List of *ChannelAdminLogEvent*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *channels.GetAdminLog*.

### ChannelParticipant

```
class pyrogram.api.types.channels.ChannelParticipant
```

**ID**

0xd0d9b163

**Parameters**

- **participant** – Either *ChannelParticipant*, *ChannelParticipantSelf*, *ChannelParticipantCreator*, *ChannelParticipantAdmin* or *ChannelParticipantBanned*

- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *channels.GetParticipant*.

**ChannelParticipants**

**class** pyrogram.api.types.channels.**ChannelParticipants**

**ID**

0xf56ee2a8

**Parameters**

- **count** – int 32-bit
- **participants** – List of either *ChannelParticipant*, *ChannelParticipantSelf*, *ChannelParticipantCreator*, *ChannelParticipantAdmin* or *ChannelParticipantBanned*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *channels.GetParticipants*.

**ChannelParticipantsNotModified**

**class** pyrogram.api.types.channels.**ChannelParticipantsNotModified**

**ID**

0xf0173fe9

No parameters required.

**See also:**

This object can be returned by *channels.GetParticipants*.

**4.37.4 Contacts****Blocked**

**class** pyrogram.api.types.contacts.**Blocked**

**ID**

0x1c138d15

**Parameters**

- **blocked** – List of *ContactBlocked*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by `contacts.GetBlocked`.

## BlockedSlice

**class** pyrogram.api.types.contacts.BlockedSlice

**ID**

0x900802a1

**Parameters**

- **count** – int 32-bit
- **blocked** – List of *ContactBlocked*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by `contacts.GetBlocked`.

## Contacts

**class** pyrogram.api.types.contacts.Contacts

**ID**

0xae87e42

**Parameters**

- **contacts** – List of *Contact*
- **saved\_count** – int 32-bit
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by `contacts.GetContacts`.

## ContactsNotModified

**class** pyrogram.api.types.contacts.ContactsNotModified

**ID**

0xb74ba9d2

No parameters required.

**See also:**

This object can be returned by `contacts.GetContacts`.

## Found

**class** pyrogram.api.types.contacts.**Found**

### ID

0xb3134d9d

### Parameters

- **my\_results** – List of either *PeerUser*, *PeerChat* or *PeerChannel*
- **results** – List of either *PeerUser*, *PeerChat* or *PeerChannel*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

### See also:

This object can be returned by *contacts.Search*.

## ImportedContacts

**class** pyrogram.api.types.contacts.**ImportedContacts**

### ID

0x77d01c3b

### Parameters

- **imported** – List of *ImportedContact*
- **popular\_invites** – List of *PopularContact*
- **retry\_contacts** – List of int 64-bit
- **users** – List of either *UserEmpty* or *User*

### See also:

This object can be returned by *contacts.ImportContacts*.

## Link

**class** pyrogram.api.types.contacts.**Link**

### ID

0x3ace484c

### Parameters

- **my\_link** – Either *ContactLinkUnknown*, *ContactLinkNone*, *ContactLinkHasPhone* or *ContactLinkContact*
- **foreign\_link** – Either *ContactLinkUnknown*, *ContactLinkNone*, *ContactLinkHasPhone* or *ContactLinkContact*

- **user** – Either *UserEmpty* or *User*

**See also:**

This object can be returned by *contacts.DeleteContact*.

## ResolvedPeer

**class** pyrogram.api.types.contacts.ResolvedPeer

**ID**

0x7f077ad9

**Parameters**

- **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *contacts.ResolveUsername*.

## TopPeers

**class** pyrogram.api.types.contacts.TopPeers

**ID**

0x70b772a8

**Parameters**

- **categories** – List of *TopPeerCategoryPeers*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *contacts.GetTopPeers*.

## TopPeersDisabled

**class** pyrogram.api.types.contacts.TopPeersDisabled

**ID**

0xb52c939d



No parameters required.

**See also:**

This object can be returned by `contacts.GetTopPeers`.

## TopPeersNotModified

**class** pyrogram.api.types.contacts.TopPeersNotModified

**ID**

0xde266ef5

No parameters required.

**See also:**

This object can be returned by `contacts.GetTopPeers`.

## 4.37.5 Help

### AppUpdate

**class** pyrogram.api.types.help.AppUpdate

**ID**

0x1da7158f

**Parameters**

- **id** – int 32-bit
- **version** – str
- **text** – str
- **entities** – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*
- **popup** (*optional*) – bool
- **document** (*optional*) – Either *DocumentEmpty* or *Document*
- **url** (*optional*) – str

**See also:**

This object can be returned by `help.GetAppUpdate`.

## ConfigSimple

**class** pyrogram.api.types.help.ConfigSimple

**ID**

0x5a592a6c

### Parameters

- **date** – int 32-bit
- **expires** – int 32-bit
- **rules** – vector

## DeepLinkInfo

**class** pyrogram.api.types.help.DeepLinkInfo

**ID**

0x6a4ee832

### Parameters

- **message** – str
- **update\_app** (*optional*) – bool
- **entities** (*optional*) – List of either *MessageEntityUnknown*,  
*MessageEntityMention*, *MessageEntityHashtag*,  
*MessageEntityBotCommand*, *MessageEntityUrl*,  
*MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*,  
*MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*,  
*MessageEntityMentionName*, *InputMessageEntityMentionName*,  
*MessageEntityPhone* or *MessageEntityCashtag*

### See also:

This object can be returned by *help.GetDeepLinkInfo*.

## DeepLinkInfoEmpty

**class** pyrogram.api.types.help.DeepLinkInfoEmpty

**ID**

0x66afa166

No parameters required.

### See also:

This object can be returned by *help.GetDeepLinkInfo*.

### InviteText

**class** pyrogram.api.types.help.InviteText

**ID**

0x18cb9f78

**Parameters** *message* – str

**See also:**

This object can be returned by [\*help.GetInviteText\*](#).

### NoAppUpdate

**class** pyrogram.api.types.help.NoAppUpdate

**ID**

0xc45a6536

No parameters required.

**See also:**

This object can be returned by [\*help.GetAppUpdate\*](#).

### PassportConfig

**class** pyrogram.api.types.help.PassportConfig

**ID**

0xa098d6af

**Parameters**

- **hash** – int 32-bit
- **countries\_langs** – *DataJSON*

**See also:**

This object can be returned by [\*help.GetPassportConfig\*](#).

### PassportConfigNotModified

**class** pyrogram.api.types.help.PassportConfigNotModified

**ID**

0xbfb9f457

No parameters required.

**See also:**

This object can be returned by [\*help.GetPassportConfig\*](#).

## ProxyDataEmpty

**class** pyrogram.api.types.help.ProxyDataEmpty

**ID**

0xe09e1fb8

**Parameters** **expires** – int 32-bit

**See also:**

This object can be returned by [\*help.GetProxyData\*](#).

## ProxyDataPromo

**class** pyrogram.api.types.help.ProxyDataPromo

**ID**

0x2bf7ee23

**Parameters**

- **expires** – int 32-bit
- **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by [\*help.GetProxyData\*](#).

## RecentMeUrls

**class** pyrogram.api.types.help.RecentMeUrls

**ID**

0x0e0310d7

**Parameters**

- **urls** – List of either *RecentMeUrlUnknown*, *RecentMeUrlUser*, *RecentMeUrlChat*, *RecentMeUrlChatInvite* or *RecentMeUrlStickerSet*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by [\*help.GetRecentMeUrls\*](#).

## Support

**class** pyrogram.api.types.help.Support

**ID**  
0x17c6b5f6

### Parameters

- **phone\_number** – str
- **user** – Either *UserEmpty* or *User*

### See also:

This object can be returned by *help.GetSupport*.

## SupportName

**class** pyrogram.api.types.help.SupportName

**ID**  
0x8c05f1c9

**Parameters** **name** – str

### See also:

This object can be returned by *help.GetSupportName*.

## TermsOfService

**class** pyrogram.api.types.help.TermsOfService

**ID**  
0x780a0310

### Parameters

- **id** – *DataJSON*
- **text** – str
- **entities** – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*
- **popup** (*optional*) – bool
- **min\_age\_confirm** (*optional*) – int 32-bit

## TermsOfServiceUpdate

**class** pyrogram.api.types.help.TermsOfServiceUpdate

**ID**

0x28ecf961

### Parameters

- **expires** – int 32-bit
- **terms\_of\_service** – [help.TermsOfService](#)

**See also:**

This object can be returned by [help.GetTermsOfServiceUpdate](#).

## TermsOfServiceUpdateEmpty

**class** pyrogram.api.types.help.TermsOfServiceUpdateEmpty

**ID**

0xe3309f7f

**Parameters** **expires** – int 32-bit

**See also:**

This object can be returned by [help.GetTermsOfServiceUpdate](#).

## UserInfo

**class** pyrogram.api.types.help.UserInfo

**ID**

0x01eb3758

### Parameters

- **message** – str
- **entities** – List of either [MessageEntityUnknown](#), [MessageEntityMention](#), [MessageEntityHashtag](#), [MessageEntityBotCommand](#), [MessageEntityUrl](#), [MessageEntityEmail](#), [MessageEntityBold](#), [MessageEntityItalic](#), [MessageEntityCode](#), [MessageEntityPre](#), [MessageEntityTextUrl](#), [MessageEntityMentionName](#), [InputMessageEntityMentionName](#), [MessageEntityPhone](#) or [MessageEntityCashtag](#)
- **author** – str
- **date** – int 32-bit

**See also:**

This object can be returned by [help.GetUserInfo](#) and [help.EditUserInfo](#).

### UserInfoEmpty

**class** pyrogram.api.types.help.UserInfoEmpty

**ID**

0xf3ae2eed

No parameters required.

**See also:**

This object can be returned by [\*help.GetUserInfo\*](#) and [\*help.EditUserInfo\*](#).

## 4.37.6 Messages

### AffectedHistory

**class** pyrogram.api.types.messages.AffectedHistory

**ID**

0xb45c69d1

**Parameters**

- **pts** – int 32-bit
- **pts\_count** – int 32-bit
- **offset** – int 32-bit

**See also:**

This object can be returned by [\*messages.DeleteHistory\*](#), [\*messages.ReadMentions\*](#) and [\*channels.DeleteUserHistory\*](#).

### AffectedMessages

**class** pyrogram.api.types.messages.AffectedMessages

**ID**

0x84d19185

**Parameters**

- **pts** – int 32-bit
- **pts\_count** – int 32-bit

**See also:**

This object can be returned by [\*messages.ReadHistory\*](#), [\*messages.DeleteMessages\*](#), [\*messages.ReadMessageContents\*](#) and [\*channels.DeleteMessages\*](#).

## AllStickers

**class** pyrogram.api.types.messages.**AllStickers**

**ID**  
0xedfd405f

### Parameters

- **hash** – int 32-bit
- **sets** – List of *StickerSet*

### See also:

This object can be returned by *messages.GetAllStickers* and *messages.GetMaskStickers*.

## AllStickersNotModified

**class** pyrogram.api.types.messages.**AllStickersNotModified**

**ID**  
0xe86602c3

No parameters required.

### See also:

This object can be returned by *messages.GetAllStickers* and *messages.GetMaskStickers*.

## ArchivedStickers

**class** pyrogram.api.types.messages.**ArchivedStickers**

**ID**  
0x4fcba9c8

### Parameters

- **count** – int 32-bit
- **sets** – List of either *StickerSetCovered* or *StickerSetMultiCovered*

### See also:

This object can be returned by *messages.GetArchivedStickers*.

## BotCallbackAnswer

**class** pyrogram.api.types.messages.**BotCallbackAnswer**

**ID**  
0x36585ea4



**Parameters**

- **cache\_time** – int 32-bit
- **alert** (*optional*) – bool
- **has\_url** (*optional*) – bool
- **native\_ui** (*optional*) – bool
- **message** (*optional*) – str
- **url** (*optional*) – str

**See also:**

This object can be returned by `messages.GetBotCallbackAnswer`.

**BotResults**

```
class pyrogram.api.types.messages.BotResults
```

**ID**

0x947ca848

**Parameters**

- **query\_id** – int 64-bit
- **results** – List of either *BotInlineResult* or *BotInlineMediaResult*
- **cache\_time** – int 32-bit
- **users** – List of either *UserEmpty* or *User*
- **gallery** (*optional*) – bool
- **next\_offset** (*optional*) – str
- **switch\_pm** (*optional*) – *InlineBotSwitchPM*

**See also:**

This object can be returned by `messages.GetInlineBotResults`.

**ChannelMessages**

```
class pyrogram.api.types.messages.ChannelMessages
```

**ID**

0x99262e37

**Parameters**

- **pts** – int 32-bit
- **count** – int 32-bit
- **messages** – List of either *MessageEmpty*, *Message* or *MessageService*

- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*
- **inexact** (*optional*) – bool

**See also:**

This object can be returned by *messages.GetMessages*, *messages.GetHistory*, *messages.Search*, *messages.SearchGlobal*, *messages.GetUnreadMentions*, *messages.GetRecentLocations* and *channels.GetMessages*.

## ChatFull

**class** pyrogram.api.types.messages.ChatFull

**ID**

0xe5d7d19c

**Parameters**

- **full\_chat** – Either *ChatFull* or *ChannelFull*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *messages.GetFullChat* and *channels.GetFullChannel*.

## Chats

**class** pyrogram.api.types.messages.Chats

**ID**

0x64ff9fd5

**Parameters** **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*

**See also:**

This object can be returned by *messages.GetChats*, *messages.GetCommonChats*, *messages.GetAllChats*, *channels.GetChannels*, *channels.GetAdmindedPublicChannels* and *channels.GetLeftChannels*.

## ChatsSlice

**class** pyrogram.api.types.messages.ChatsSlice

**ID**

0x9cd81144

**Parameters**

- **count** – int 32-bit
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*

**See also:**

This object can be returned by *messages.GetChats*, *messages.GetCommonChats*, *messages.GetAllChats*, *channels.GetChannels*, *channels.GetAdminedPublicChannels* and *channels.GetLeftChannels*.

**DhConfig**

```
class pyrogram.api.types.messages.DhConfig
```

**ID**

0x2c221edd

**Parameters**

- **g** – int 32-bit
- **p** – bytes
- **version** – int 32-bit
- **random** – bytes

**See also:**

This object can be returned by *messages.GetDhConfig*.

**DhConfigNotModified**

```
class pyrogram.api.types.messages.DhConfigNotModified
```

**ID**

0xc0e24635

**Parameters** **random** – bytes

**See also:**

This object can be returned by *messages.GetDhConfig*.

## Dialogs

**class** pyrogram.api.types.messages.Dialogs

**ID**

0x15ba6c40

### Parameters

- **dialogs** – List of *Dialog*
- **messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *messages.GetDialogs*.

## DialogsNotModified

**class** pyrogram.api.types.messages.DialogsNotModified

**ID**

0xf0e3e596

**Parameters** **count** – int 32-bit

**See also:**

This object can be returned by *messages.GetDialogs*.

## DialogsSlice

**class** pyrogram.api.types.messages.DialogsSlice

**ID**

0x71e094f3

### Parameters

- **count** – int 32-bit
- **dialogs** – List of *Dialog*
- **messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by `messages.GetDialogs`.

**FavedStickers**

```
class pyrogram.api.types.messages.FavedStickers
```

**ID**

0xf37f2f16

**Parameters**

- **hash** – int 32-bit
- **packs** – List of *StickerPack*
- **stickers** – List of either *DocumentEmpty* or *Document*

**See also:**

This object can be returned by `messages.GetFavedStickers`.

**FavedStickersNotModified**

```
class pyrogram.api.types.messages.FavedStickersNotModified
```

**ID**

0x9e8fa6d3

No parameters required.

**See also:**

This object can be returned by `messages.GetFavedStickers`.

**FeaturedStickers**

```
class pyrogram.api.types.messages.FeaturedStickers
```

**ID**

0xf89d88e5

**Parameters**

- **hash** – int 32-bit
- **sets** – List of either *StickerSetCovered* or *StickerSetMultiCovered*
- **unread** – List of int 64-bit

**See also:**

This object can be returned by `messages.GetFeaturedStickers`.

### FeaturedStickersNotModified

**class** pyrogram.api.types.messages.FeaturedStickersNotModified

**ID**

0x04ede3cf

No parameters required.

**See also:**

This object can be returned by *messages.GetFeaturedStickers*.

### FoundGifs

**class** pyrogram.api.types.messages.FoundGifs

**ID**

0x450a1c0a

**Parameters**

- **next\_offset** – int 32-bit
- **results** – List of either *FoundGif* or *FoundGifCached*

**See also:**

This object can be returned by *messages.SearchGifs*.

### FoundStickerSets

**class** pyrogram.api.types.messages.FoundStickerSets

**ID**

0x5108d648

**Parameters**

- **hash** – int 32-bit
- **sets** – List of either *StickerSetCovered* or *StickerSetMultiCovered*

**See also:**

This object can be returned by *messages.SearchStickerSets*.

### FoundStickerSetsNotModified

**class** pyrogram.api.types.messages.FoundStickerSetsNotModified

**ID**

0x0d54b65d

No parameters required.

**See also:**

This object can be returned by `messages.SearchStickerSets`.

## HighScores

**class** `pyrogram.api.types.messages.HighScores`

**ID**

0x9a3bfd99

**Parameters**

- **scores** – List of *HighScore*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by `messages.GetGameHighScores` and `messages.GetInlineGameHighScores`.

## MessageEditData

**class** `pyrogram.api.types.messages.MessageEditData`

**ID**

0x26b5dde6

**Parameters** **caption** (*optional*) – bool

**See also:**

This object can be returned by `messages.GetMessageEditData`.

## Messages

**class** `pyrogram.api.types.messages.Messages`

**ID**

0x8c718e87

**Parameters**

- **messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *messages.GetMessages*, *messages.GetHistory*, *messages.Search*, *messages.SearchGlobal*, *messages.GetUnreadMentions*, *messages.GetRecentLocations* and *channels.GetMessages*.

## MessagesNotModified

**class** pyrogram.api.types.messages.**MessagesNotModified**

**ID**

0x74535f21

**Parameters** **count** – int 32-bit

**See also:**

This object can be returned by *messages.GetMessages*, *messages.GetHistory*, *messages.Search*, *messages.SearchGlobal*, *messages.GetUnreadMentions*, *messages.GetRecentLocations* and *channels.GetMessages*.

## MessagesSlice

**class** pyrogram.api.types.messages.**MessagesSlice**

**ID**

0xa6c47aaa

**Parameters**

- **count** – int 32-bit
- **messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*
- **inexact** (*optional*) – bool

**See also:**

This object can be returned by *messages.GetMessages*, *messages.GetHistory*, *messages.Search*, *messages.SearchGlobal*, *messages.GetUnreadMentions*, *messages.GetRecentLocations* and *channels.GetMessages*.

## PeerDialogs

**class** pyrogram.api.types.messages.**PeerDialogs**

**ID**

0x3371c354



**Parameters**

- **dialogs** – List of *Dialog*
- **messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*
- **state** – *updates.State*

**See also:**

This object can be returned by *messages.GetPeerDialogs* and *messages.GetPinnedDialogs*.

**RecentStickers**

```
class pyrogram.api.types.messages.RecentStickers
```

**ID**

0x22f3afb3

**Parameters**

- **hash** – int 32-bit
- **packs** – List of *StickerPack*
- **stickers** – List of either *DocumentEmpty* or *Document*
- **dates** – List of int 32-bit

**See also:**

This object can be returned by *messages.GetRecentStickers*.

**RecentStickersNotModified**

```
class pyrogram.api.types.messages.RecentStickersNotModified
```

**ID**

0x0b17f890

No parameters required.

**See also:**

This object can be returned by *messages.GetRecentStickers*.

**SavedGifs**

```
class pyrogram.api.types.messages.SavedGifs
```

**ID**

0x2e0709a5

#### Parameters

- **hash** – int 32-bit
- **gifs** – List of either *DocumentEmpty* or *Document*

#### See also:

This object can be returned by *messages.GetSavedGifs*.

### SavedGifsNotModified

**class** pyrogram.api.types.messages.SavedGifsNotModified

#### ID

0xe8025ca2

No parameters required.

#### See also:

This object can be returned by *messages.GetSavedGifs*.

### SentEncryptedFile

**class** pyrogram.api.types.messages.SentEncryptedFile

#### ID

0x9493ff32

#### Parameters

- **date** – int 32-bit
- **file** – Either *EncryptedFileEmpty* or *EncryptedFile*

#### See also:

This object can be returned by *messages.SendEncrypted*, *messages.SendEncryptedFile* and *messages.SendEncryptedService*.

### SentEncryptedMessage

**class** pyrogram.api.types.messages.SentEncryptedMessage

#### ID

0x560f8935

**Parameters** **date** – int 32-bit

#### See also:

This object can be returned by *messages.SendEncrypted*, *messages.SendEncryptedFile* and *messages.SendEncryptedService*.

## StickerSet

**class** pyrogram.api.types.messages.StickerSet

**ID**

0xb60a24a6

**Parameters**

- **set** – *StickerSet*
- **packs** – List of *StickerPack*
- **documents** – List of either *DocumentEmpty* or *Document*

**See also:**

This object can be returned by *messages.GetStickerSet*, *stickers.CreateStickerSet*, *stickers.RemoveStickerFromSet*, *stickers.ChangeStickerPosition* and *stickers.AddStickerToSet*.

## StickerSetInstallResultArchive

**class** pyrogram.api.types.messages.StickerSetInstallResultArchive

**ID**

0x35e410a8

**Parameters** **sets** – List of either *StickerSetCovered* or *StickerSetMultiCovered*

**See also:**

This object can be returned by *messages.InstallStickerSet*.

## StickerSetInstallResultSuccess

**class** pyrogram.api.types.messages.StickerSetInstallResultSuccess

**ID**

0x38641628

No parameters required.

**See also:**

This object can be returned by *messages.InstallStickerSet*.

## Stickers

**class** pyrogram.api.types.messages.Stickers

**ID**

0xe4599bbd

**Parameters**

- **hash** – int 32-bit
- **stickers** – List of either *DocumentEmpty* or *Document*

**See also:**

This object can be returned by *messages.GetStickers*.

**StickersNotModified**

```
class pyrogram.api.types.messages.StickersNotModified
```

**ID**

0xf1749a22

No parameters required.

**See also:**

This object can be returned by *messages.GetStickers*.

## 4.37.7 Payments

**PaymentForm**

```
class pyrogram.api.types.payments.PaymentForm
```

**ID**

0x3f56aea3

**Parameters**

- **bot\_id** – int 32-bit
- **invoice** – *Invoice*
- **provider\_id** – int 32-bit
- **url** – str
- **users** – List of either *UserEmpty* or *User*
- **can\_save\_credentials** (*optional*) – bool
- **password\_missing** (*optional*) – bool
- **native\_provider** (*optional*) – str
- **native\_params** (*optional*) – *DataJSON*
- **saved\_info** (*optional*) – *PaymentRequestedInfo*
- **saved\_credentials** (*optional*) – *PaymentSavedCredentialsCard*

**See also:**

This object can be returned by *payments.GetPaymentForm*.

## PaymentReceipt

```
class pyrogram.api.types.payments.PaymentReceipt
```

ID

0x500911e1

### Parameters

- **date** – int 32-bit
- **bot\_id** – int 32-bit
- **invoice** – *Invoice*
- **provider\_id** – int 32-bit
- **currency** – str
- **total\_amount** – int 64-bit
- **credentials\_title** – str
- **users** – List of either *UserEmpty* or *User*
- **info** (*optional*) – *PaymentRequestedInfo*
- **shipping** (*optional*) – *ShippingOption*

See also:

This object can be returned by *payments.GetPaymentReceipt*.

## PaymentResult

```
class pyrogram.api.types.payments.PaymentResult
```

ID

0x4e5f810d

**Parameters updates** – Either *UpdatesTooLong*, *UpdateShortMessage*, *UpdateShortChatMessage*, *UpdateShort*, *UpdatesCombined*, *Update* or *UpdateShortSentMessage*

See also:

This object can be returned by *payments.SendPaymentForm*.

## PaymentVerificationNeeded

```
class pyrogram.api.types.payments.PaymentVerificationNeeded
```

ID

0x6b56b921

**Parameters url** – str

**See also:**

This object can be returned by `payments.SendPaymentForm`.

**SavedInfo**

```
class pyrogram.api.types.payments.SavedInfo
```

**ID**

0xfb8fe43c

**Parameters**

- **has\_saved\_credentials** (*optional*) – bool
- **saved\_info** (*optional*) – *PaymentRequestedInfo*

**See also:**

This object can be returned by `payments.GetSavedInfo`.

**ValidatedRequestedInfo**

```
class pyrogram.api.types.payments.ValidatedRequestedInfo
```

**ID**

0xd1451883

**Parameters**

- **id** (*optional*) – str
- **shipping\_options** (*optional*) – List of *ShippingOption*

**See also:**

This object can be returned by `payments.ValidateRequestedInfo`.

## 4.37.8 Phone

**PhoneCall**

```
class pyrogram.api.types.phone.PhoneCall
```

**ID**

0xec82e140

**Parameters**

- **phone\_call** – Either *PhoneCallEmpty*, *PhoneCallWaiting*, *PhoneCallRequested*, *PhoneCallAccepted*, *PhoneCall* or *PhoneCallDiscarded*
- **users** – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by `phone.RequestCall`, `phone.AcceptCall` and `phone.ConfirmCall`.

## 4.37.9 Photos

### Photo

**class** pyrogram.api.types.photos.**Photo**

**ID**

0x20212ca8

**Parameters**

- **photo** – Either `PhotoEmpty` or `Photo`
- **users** – List of either `UserEmpty` or `User`

**See also:**

This object can be returned by `photos.UploadProfilePhoto`.

### Photos

**class** pyrogram.api.types.photos.**Photos**

**ID**

0x8dca6aa5

**Parameters**

- **photos** – List of either `PhotoEmpty` or `Photo`
- **users** – List of either `UserEmpty` or `User`

**See also:**

This object can be returned by `photos.GetUserPhotos`.

### PhotosSlice

**class** pyrogram.api.types.photos.**PhotosSlice**

**ID**

0x15051f54

**Parameters**

- **count** – int 32-bit
- **photos** – List of either `PhotoEmpty` or `Photo`
- **users** – List of either `UserEmpty` or `User`

See also:

This object can be returned by `photos.GetUserPhotos`.

### 4.37.10 Storage

#### FileGif

```
class pyrogram.api.types.storage.FileGif
```

**ID**

0xcae1aadf

No parameters required.

#### FileJpeg

```
class pyrogram.api.types.storage.FileJpeg
```

**ID**

0x007efe0e

No parameters required.

#### FileMov

```
class pyrogram.api.types.storage.FileMov
```

**ID**

0x4b09ebbc

No parameters required.

#### FileMp3

```
class pyrogram.api.types.storage.FileMp3
```

**ID**

0x528a0677

No parameters required.

#### FileMp4

```
class pyrogram.api.types.storage.FileMp4
```

**ID**

0xb3cea0e4

No parameters required.



### FilePartial

```
class pyrogram.api.types.storage.FilePartial
```

**ID**

0x40bc6f52

No parameters required.

### FilePdf

```
class pyrogram.api.types.storage.FilePdf
```

**ID**

0xae1e508d

No parameters required.

### FilePng

```
class pyrogram.api.types.storage.FilePng
```

**ID**

0x0a4f63c0

No parameters required.

### FileUnknown

```
class pyrogram.api.types.storage.FileUnknown
```

**ID**

0xaa963b05

No parameters required.

### FileWebp

```
class pyrogram.api.types.storage.FileWebp
```

**ID**

0x1081464c

No parameters required.

## 4.37.11 Updates

### ChannelDifference

**class** pyrogram.api.types.updates.**ChannelDifference**

**ID**

0x2064674e

#### Parameters

- **pts** – int 32-bit
- **new\_messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **other\_updates** – List of either *UpdateNewMessage*, *UpdateMessageID*, *UpdateDeleteMessages*, *UpdateUserTyping*, *UpdateChatUserTyping*, *UpdateChatParticipants*, *UpdateUserStatus*, *UpdateUserName*, *UpdateUserPhoto*, *UpdateContactLink*, *UpdateNewEncryptedMessage*, *UpdateEncryptedChatTyping*, *UpdateEncryption*, *UpdateEncryptedMessagesRead*, *UpdateChatParticipantAdd*, *UpdateChatParticipantDelete*, *UpdateDcOptions*, *UpdateUserBlocked*, *UpdateNotifySettings*, *UpdateServiceNotification*, *UpdatePrivacy*, *UpdateUserPhone*, *UpdateReadHistoryInbox*, *UpdateReadHistoryOutbox*, *UpdateWebPage*, *UpdateReadMessagesContents*, *UpdateChannelTooLong*, *UpdateChannel*, *UpdateNewChannelMessage*, *UpdateReadChannelInbox*, *UpdateDeleteChannelMessages*, *UpdateChannelMessageViews*, *UpdateChatParticipantAdmin*, *UpdateNewStickerSet*, *UpdateStickerSetsOrder*, *UpdateStickerSets*, *UpdateSavedGifs*, *UpdateBotInlineQuery*, *UpdateBotInlineSend*, *UpdateEditChannelMessage*, *UpdateChannelPinnedMessage*, *UpdateBotCallbackQuery*, *UpdateEditMessage*, *UpdateInlineBotCallbackQuery*, *UpdateReadChannelOutbox*, *UpdateDraftMessage*, *UpdateReadFeaturedStickers*, *UpdateRecentStickers*, *UpdateConfig*, *UpdatePtsChanged*, *UpdateChannelWebPage*, *UpdateDialogPinned*, *UpdatePinnedDialogs*, *UpdateBotWebhookJSON*, *UpdateBotWebhookJSONQuery*, *UpdateBotShippingQuery*, *UpdateBotPrecheckoutQuery*, *UpdatePhoneCall*, *UpdateLangPackTooLong*, *UpdateLangPack*, *UpdateFavedStickers*, *UpdateChannelReadMessagesContents*, *UpdateContactsReset*, *UpdateChannelAvailableMessages*, *UpdateDialogUnreadMark*, *UpdateUserPinnedMessage*, *UpdateChatPinnedMessage*, *UpdateMessagePoll* or *UpdateChatDefaultBannedRights*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*
- **final** (*optional*) – bool
- **timeout** (*optional*) – int 32-bit

See also:

This object can be returned by `updates.GetChannelDifference`.

### ChannelDifferenceEmpty

**class** pyrogram.api.types.updates.ChannelDifferenceEmpty

**ID**

0x3e11affb

#### Parameters

- **pts** – int 32-bit
- **final** (*optional*) – bool
- **timeout** (*optional*) – int 32-bit

**See also:**

This object can be returned by `updates.GetChannelDifference`.

### ChannelDifferenceTooLong

**class** pyrogram.api.types.updates.ChannelDifferenceTooLong

**ID**

0x6a9d7b35

#### Parameters

- **pts** – int 32-bit
- **top\_message** – int 32-bit
- **read\_inbox\_max\_id** – int 32-bit
- **read\_outbox\_max\_id** – int 32-bit
- **unread\_count** – int 32-bit
- **unread\_mentions\_count** – int 32-bit
- **messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*
- **final** (*optional*) – bool
- **timeout** (*optional*) – int 32-bit

**See also:**

This object can be returned by `updates.GetChannelDifference`.

## Difference

**class** pyrogram.api.types.updates.**Difference**

**ID**

0x00f49ca0

### Parameters

- **new\_messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **new\_encrypted\_messages** – List of either *EncryptedMessage* or *EncryptedMessageService*
- **other\_updates** – List of either *UpdateNewMessage*, *UpdateMessageID*, *UpdateDeleteMessages*, *UpdateUserTyping*, *UpdateChatUserTyping*, *UpdateChatParticipants*, *UpdateUserStatus*, *UpdateUserName*, *UpdateUserPhoto*, *UpdateContactLink*, *UpdateNewEncryptedMessage*, *UpdateEncryptedChatTyping*, *UpdateEncryption*, *UpdateEncryptedMessagesRead*, *UpdateChatParticipantAdd*, *UpdateChatParticipantDelete*, *UpdateDcOptions*, *UpdateUserBlocked*, *UpdateNotifySettings*, *UpdateServiceNotification*, *UpdatePrivacy*, *UpdateUserPhone*, *UpdateReadHistoryInbox*, *UpdateReadHistoryOutbox*, *UpdateWebPage*, *UpdateReadMessagesContents*, *UpdateChannelTooLong*, *UpdateChannel*, *UpdateNewChannelMessage*, *UpdateReadChannelInbox*, *UpdateDeleteChannelMessages*, *UpdateChannelMessageViews*, *UpdateChatParticipantAdmin*, *UpdateNewStickerSet*, *UpdateStickerSetsOrder*, *UpdateStickerSets*, *UpdateSavedGifs*, *UpdateBotInlineQuery*, *UpdateBotInlineSend*, *UpdateEditChannelMessage*, *UpdateChannelPinnedMessage*, *UpdateBotCallbackQuery*, *UpdateEditMessage*, *UpdateInlineBotCallbackQuery*, *UpdateReadChannelOutbox*, *UpdateDraftMessage*, *UpdateReadFeaturedStickers*, *UpdateRecentStickers*, *UpdateConfig*, *UpdatePtsChanged*, *UpdateChannelWebPage*, *UpdateDialogPinned*, *UpdatePinnedDialogs*, *UpdateBotWebhookJSON*, *UpdateBotWebhookJSONQuery*, *UpdateBotShippingQuery*, *UpdateBotPrecheckoutQuery*, *UpdatePhoneCall*, *UpdateLangPackTooLong*, *UpdateLangPack*, *UpdateFavedStickers*, *UpdateChannelReadMessagesContents*, *UpdateContactsReset*, *UpdateChannelAvailableMessages*, *UpdateDialogUnreadMark*, *UpdateUserPinnedMessage*, *UpdateChatPinnedMessage*, *UpdateMessagePoll* or *UpdateChatDefaultBannedRights*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*
- **state** – *updates.State*

**See also:**

This object can be returned by *updates.GetDifference*.

## DifferenceEmpty

**class** pyrogram.api.types.updates.DifferenceEmpty

**ID**

0x5d75a138

### Parameters

- **date** – int 32-bit
- **seq** – int 32-bit

**See also:**

This object can be returned by `updates.GetDifference`.

## DifferenceSlice

**class** pyrogram.api.types.updates.DifferenceSlice

**ID**

0xa8fb1981

### Parameters

- **new\_messages** – List of either *MessageEmpty*, *Message* or *MessageService*
- **new\_encrypted\_messages** – List of either *EncryptedMessage* or *EncryptedMessageService*
- **other\_updates** – List of either *UpdateNewMessage*, *UpdateMessageID*, *UpdateDeleteMessages*, *UpdateUserTyping*, *UpdateChatUserTyping*, *UpdateChatParticipants*, *UpdateUserStatus*, *UpdateUserName*, *UpdateUserPhoto*, *UpdateContactLink*, *UpdateNewEncryptedMessage*, *UpdateEncryptedChatTyping*, *UpdateEncryption*, *UpdateEncryptedMessagesRead*, *UpdateChatParticipantAdd*, *UpdateChatParticipantDelete*, *UpdateDcOptions*, *UpdateUserBlocked*, *UpdateNotifySettings*, *UpdateServiceNotification*, *UpdatePrivacy*, *UpdateUserPhone*, *UpdateReadHistoryInbox*, *UpdateReadHistoryOutbox*, *UpdateWebPage*, *UpdateReadMessagesContents*, *UpdateChannelTooLong*, *UpdateChannel*, *UpdateNewChannelMessage*, *UpdateReadChannelInbox*, *UpdateDeleteChannelMessages*, *UpdateChannelMessageViews*, *UpdateChatParticipantAdmin*, *UpdateNewStickerSet*, *UpdateStickerSetsOrder*, *UpdateStickerSets*, *UpdateSavedGifs*, *UpdateBotInlineQuery*, *UpdateBotInlineSend*, *UpdateEditChannelMessage*, *UpdateChannelPinnedMessage*, *UpdateBotCallbackQuery*, *UpdateEditMessage*, *UpdateInlineBotCallbackQuery*, *UpdateReadChannelOutbox*, *UpdateDraftMessage*, *UpdateReadFeaturedStickers*, *UpdateRecentStickers*, *UpdateConfig*, *UpdatePtsChanged*, *UpdateChannelWebPage*, *UpdateDialogPinned*, *UpdatePinnedDialogs*, *UpdateBotWebhookJSON*, *UpdateBotWebhookJSONQuery*, *UpdateBotShippingQuery*, *UpdateBotPrecheckoutQuery*,

```
UpdatePhoneCall, UpdateLangPackTooLong, UpdateLangPack,
UpdateFavedStickers, UpdateChannelReadMessagesContents,
UpdateContactsReset, UpdateChannelAvailableMessages,
UpdateDialogUnreadMark, UpdateUserPinnedMessage,
UpdateChatPinnedMessage, UpdateMessagePoll or
UpdateChatDefaultBannedRights
```

- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **users** – List of either *UserEmpty* or *User*
- **intermediate\_state** – *updates.State*

See also:

This object can be returned by *updates.GetDifference*.

## DifferenceTooLong

```
class pyrogram.api.types.updates.DifferenceTooLong
```

ID

0x4afe8f6d

**Parameters** **pts** – int 32-bit

See also:

This object can be returned by *updates.GetDifference*.

## State

```
class pyrogram.api.types.updates.State
```

ID

0xa56c2a3e

**Parameters**

- **pts** – int 32-bit
- **qts** – int 32-bit
- **date** – int 32-bit
- **seq** – int 32-bit
- **unread\_count** – int 32-bit

See also:

This object can be returned by *updates.GetState*.

## 4.37.12 Upload

### CdnFile

**class** pyrogram.api.types.upload.CdnFile

**ID**

0xa99fca4f

**Parameters** **bytes** – bytes

**See also:**

This object can be returned by [\*upload.GetCdnFile\*](#).

### CdnFileReuploadNeeded

**class** pyrogram.api.types.upload.CdnFileReuploadNeeded

**ID**

0xeea8e46e

**Parameters** **request\_token** – bytes

**See also:**

This object can be returned by [\*upload.GetCdnFile\*](#).

### File

**class** pyrogram.api.types.upload.File

**ID**

0x096a18d5

**Parameters**

- **type** – Either [\*storage.FileUnknown\*](#), [\*storage.FilePartial\*](#), [\*storage.FileJpeg\*](#), [\*storage.FileGif\*](#), [\*storage.FilePng\*](#), [\*storage.FilePdf\*](#), [\*storage.FileMp3\*](#), [\*storage.FileMov\*](#), [\*storage.FileMp4\*](#) or [\*storage.FileWebp\*](#)
- **mtime** – int 32-bit
- **bytes** – bytes

**See also:**

This object can be returned by [\*upload.GetFile\*](#).

## FileCdnRedirect

**class** pyrogram.api.types.upload.**FileCdnRedirect**

**ID**

0xf18cda44

### Parameters

- **dc\_id** – int 32-bit
- **file\_token** – bytes
- **encryption\_key** – bytes
- **encryption\_iv** – bytes
- **file\_hashes** – List of *FileHash*

**See also:**

This object can be returned by *upload.GetFile*.

## WebFile

**class** pyrogram.api.types.upload.**WebFile**

**ID**

0x21e753bc

### Parameters

- **size** – int 32-bit
- **mime\_type** – str
- **file\_type** – Either *storage.FileUnknown*, *storage.FilePartial*, *storage.FileJpeg*, *storage.FileGif*, *storage.FilePng*, *storage.FilePdf*, *storage.FileMp3*, *storage.FileMov*, *storage.FileMp4* or *storage.FileWebp*
- **mtime** – int 32-bit
- **bytes** – bytes

**See also:**

This object can be returned by *upload.GetWebFile*.

## 4.37.13 AccessPointRule

**class** pyrogram.api.types.**AccessPointRule**

**ID**

0x4679b65f

### Parameters



- **phone\_prefix\_rules** – str
- **dc\_id** – int 32-bit
- **ips** – vector

#### 4.37.14 AccountDaysTTL

**class** pyrogram.api.types.AccountDaysTTL

**ID**

0xb8d0afdf

**Parameters** **days** – int 32-bit

**See also:**

This object can be returned by [\*account.GetAccountTTL\*](#).

#### 4.37.15 Authorization

**class** pyrogram.api.types.Authorization

**ID**

0xad01d61d

**Parameters**

- **hash** – int 64-bit
- **device\_model** – str
- **platform** – str
- **system\_version** – str
- **api\_id** – int 32-bit
- **app\_name** – str
- **app\_version** – str
- **date\_created** – int 32-bit
- **date\_active** – int 32-bit
- **ip** – str
- **country** – str
- **region** – str
- **current** (*optional*) – bool
- **official\_app** (*optional*) – bool
- **password\_pending** (*optional*) – bool

### 4.37.16 AutoDownloadSettings

**class** pyrogram.api.types.AutoDownloadSettings

**ID**

0xd246fd47

**Parameters**

- **photo\_size\_max** – int 32-bit
- **video\_size\_max** – int 32-bit
- **file\_size\_max** – int 32-bit
- **disabled** (*optional*) – bool
- **video\_preload\_large** (*optional*) – bool
- **audio\_preload\_next** (*optional*) – bool
- **phonecalls\_less\_data** (*optional*) – bool

### 4.37.17 BadMsgNotification

**class** pyrogram.api.types.BadMsgNotification

**ID**

0xa7eff811

**Parameters**

- **bad\_msg\_id** – int 64-bit
- **bad\_msg\_seqno** – int 32-bit
- **error\_code** – int 32-bit

### 4.37.18 BadServerSalt

**class** pyrogram.api.types.BadServerSalt

**ID**

0xedab447b

**Parameters**

- **bad\_msg\_id** – int 64-bit
- **bad\_msg\_seqno** – int 32-bit
- **error\_code** – int 32-bit
- **new\_server\_salt** – int 64-bit

#### 4.37.19 BindAuthKeyInner

**class** pyrogram.api.types.BindAuthKeyInner

**ID**  
0x75a3f765

##### Parameters

- **nonce** – int 64-bit
- **temp\_auth\_key\_id** – int 64-bit
- **perm\_auth\_key\_id** – int 64-bit
- **temp\_session\_id** – int 64-bit
- **expires\_at** – int 32-bit

#### 4.37.20 BotCommand

**class** pyrogram.api.types.BotCommand

**ID**  
0xc27ac8c7

##### Parameters

- **command** – str
- **description** – str

#### 4.37.21 BotInfo

**class** pyrogram.api.types.BotInfo

**ID**  
0x98e81d3a

##### Parameters

- **user\_id** – int 32-bit
- **description** – str
- **commands** – List of *BotCommand*

#### 4.37.22 BotInlineMediaResult

**class** pyrogram.api.types.BotInlineMediaResult

**ID**  
0x17db940b

**Parameters**

- **id** – str
- **type** – str
- **send\_message** – Either *BotInlineMessageMediaAuto*, *BotInlineMessageText*, *BotInlineMessageMediaGeo*, *BotInlineMessageMediaVenue* or *BotInlineMessageMediaContact*
- **photo** (*optional*) – Either *PhotoEmpty* or *Photo*
- **document** (*optional*) – Either *DocumentEmpty* or *Document*
- **title** (*optional*) – str
- **description** (*optional*) – str

### 4.37.23 BotInlineMessageMediaAuto

```
class pyrogram.api.types.BotInlineMessageMediaAuto
```

**ID**

0x764cf810

**Parameters**

- **message** – str
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*
- **reply\_markup** (*optional*) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

### 4.37.24 BotInlineMessageMediaContact

```
class pyrogram.api.types.BotInlineMessageMediaContact
```

**ID**

0x18d1cdc2

**Parameters**

- **phone\_number** – str
- **first\_name** – str
- **last\_name** – str
- **vcard** – str

- **reply\_markup** (optional) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

#### 4.37.25 BotInlineMessageMediaGeo

**class** pyrogram.api.types.**BotInlineMessageMediaGeo**

**ID**

0xb722de65

**Parameters**

- **geo** – Either *GeoPointEmpty* or *GeoPoint*
- **period** – int 32-bit
- **reply\_markup** (optional) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

#### 4.37.26 BotInlineMessageMediaVenue

**class** pyrogram.api.types.**BotInlineMessageMediaVenue**

**ID**

0x8a86659c

**Parameters**

- **geo** – Either *GeoPointEmpty* or *GeoPoint*
- **title** – str
- **address** – str
- **provider** – str
- **venue\_id** – str
- **venue\_type** – str
- **reply\_markup** (optional) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

#### 4.37.27 BotInlineMessageText

**class** pyrogram.api.types.**BotInlineMessageText**

**ID**

0x8c7f65e2

**Parameters**

- **message** – str
- **no\_webpage** (*optional*) – bool
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*
- **reply\_markup** (*optional*) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

#### 4.37.28 BotInlineResult

**class** pyrogram.api.types.**BotInlineResult**

**ID**

0x11965f3a

**Parameters**

- **id** – str
- **type** – str
- **send\_message** – Either *BotInlineMessageMediaAuto*, *BotInlineMessageText*, *BotInlineMessageMediaGeo*, *BotInlineMessageMediaVenue* or *BotInlineMessageMediaContact*
- **title** (*optional*) – str
- **description** (*optional*) – str
- **url** (*optional*) – str
- **thumb** (*optional*) – Either *WebDocument* or *WebDocumentNoProxy*
- **content** (*optional*) – Either *WebDocument* or *WebDocumentNoProxy*

#### 4.37.29 CdnConfig

**class** pyrogram.api.types.**CdnConfig**

**ID**

0x5725e40a

**Parameters** **public\_keys** – List of *CdnPublicKey*

**See also:**

This object can be returned by [\*help.GetCdnConfig\*](#).

### 4.37.30 CdnPublicKey

**class** pyrogram.api.types.CdnPublicKey

**ID**

0xc982eaba

**Parameters**

- **dc\_id** – int 32-bit
- **public\_key** – str

### 4.37.31 Channel

**class** pyrogram.api.types.Channel

**ID**

0x4df30834

**Parameters**

- **id** – int 32-bit
- **title** – str
- **photo** – Either *ChatPhotoEmpty* or *ChatPhoto*
- **date** – int 32-bit
- **version** – int 32-bit
- **creator** (*optional*) – bool
- **left** (*optional*) – bool
- **broadcast** (*optional*) – bool
- **verified** (*optional*) – bool
- **megagroup** (*optional*) – bool
- **restricted** (*optional*) – bool
- **signatures** (*optional*) – bool
- **min** (*optional*) – bool
- **access\_hash** (*optional*) – int 64-bit
- **username** (*optional*) – str
- **restriction\_reason** (*optional*) – str
- **admin\_rights** (*optional*) – *ChatAdminRights*
- **banned\_rights** (*optional*) – *ChatBannedRights*
- **default\_banned\_rights** (*optional*) – *ChatBannedRights*
- **participants\_count** (*optional*) – int 32-bit

### 4.37.32 ChannelAdminLogEvent

**class** pyrogram.api.types.ChannelAdminLogEvent

**ID**

0x3b5a3e40

**Parameters**

- **id** – int 64-bit
- **date** – int 32-bit
- **user\_id** – int 32-bit
- **action** – Either *ChannelAdminLogEventActionChangeTitle*, *ChannelAdminLogEventActionChangeAbout*, *ChannelAdminLogEventActionChangeUsername*, *ChannelAdminLogEventActionChangePhoto*, *ChannelAdminLogEventActionToggleInvites*, *ChannelAdminLogEventActionToggleSignatures*, *ChannelAdminLogEventActionUpdatePinned*, *ChannelAdminLogEventActionEditMessage*, *ChannelAdminLogEventActionDeleteMessage*, *ChannelAdminLogEventActionParticipantJoin*, *ChannelAdminLogEventActionParticipantLeave*, *ChannelAdminLogEventActionParticipantInvite*, *ChannelAdminLogEventActionParticipantToggleBan*, *ChannelAdminLogEventActionParticipantToggleAdmin*, *ChannelAdminLogEventActionChangeStickerSet*, *ChannelAdminLogEventActionTogglePreHistoryHidden*, *ChannelAdminLogEventActionDefaultBannedRights* or *ChannelAdminLogEventActionStopPoll*

### 4.37.33 ChannelAdminLogEventActionChangeAbout

**class** pyrogram.api.types.ChannelAdminLogEventActionChangeAbout

**ID**

0x55188a2e

**Parameters**

- **prev\_value** – str
- **new\_value** – str

### 4.37.34 ChannelAdminLogEventActionChangePhoto

**class** pyrogram.api.types.ChannelAdminLogEventActionChangePhoto

**ID**

0xb82f55c3

**Parameters**

- **prev\_photo** – Either *ChatPhotoEmpty* or *ChatPhoto*



- **new\_photo** – Either *ChatPhotoEmpty* or *ChatPhoto*

#### 4.37.35 ChannelAdminLogEventActionChangeStickerSet

**class** pyrogram.api.types.ChannelAdminLogEventActionChangeStickerSet

**ID**

0xb1c3caa7

**Parameters**

- **prev\_stickerset** – Either *InputStickerSetEmpty*, *InputStickerSetID* or *InputStickerSetShortName*
- **new\_stickerset** – Either *InputStickerSetEmpty*, *InputStickerSetID* or *InputStickerSetShortName*

#### 4.37.36 ChannelAdminLogEventActionChangeTitle

**class** pyrogram.api.types.ChannelAdminLogEventActionChangeTitle

**ID**

0xe6dfb825

**Parameters**

- **prev\_value** – str
- **new\_value** – str

#### 4.37.37 ChannelAdminLogEventActionChangeUsername

**class** pyrogram.api.types.ChannelAdminLogEventActionChangeUsername

**ID**

0x6a4afc38

**Parameters**

- **prev\_value** – str
- **new\_value** – str

#### 4.37.38 ChannelAdminLogEventActionDefaultBannedRights

**class** pyrogram.api.types.ChannelAdminLogEventActionDefaultBannedRights

**ID**

0x2df5fc0a

**Parameters**

- **prev\_banned\_rights** – *ChatBannedRights*
- **new\_banned\_rights** – *ChatBannedRights*

#### 4.37.39 ChannelAdminLogEventActionDeleteMessage

**class** pyrogram.api.types.ChannelAdminLogEventActionDeleteMessage

**ID**

0x42e047bb

**Parameters** **message** – Either *MessageEmpty*, *Message* or *MessageService*

#### 4.37.40 ChannelAdminLogEventActionEditMessage

**class** pyrogram.api.types.ChannelAdminLogEventActionEditMessage

**ID**

0x709b2405

**Parameters**

- **prev\_message** – Either *MessageEmpty*, *Message* or *MessageService*
- **new\_message** – Either *MessageEmpty*, *Message* or *MessageService*

#### 4.37.41 ChannelAdminLogEventActionParticipantInvite

**class** pyrogram.api.types.ChannelAdminLogEventActionParticipantInvite

**ID**

0xe31c34d8

**Parameters** **participant** – Either *ChannelParticipant*,  
*ChannelParticipantSelf*, *ChannelParticipantCreator*,  
*ChannelParticipantAdmin* or *ChannelParticipantBanned*

#### 4.37.42 ChannelAdminLogEventActionParticipantJoin

**class** pyrogram.api.types.ChannelAdminLogEventActionParticipantJoin

**ID**

0x183040d3

No parameters required.

#### 4.37.43 ChannelAdminLogEventActionParticipantLeave

`class pyrogram.api.types.ChannelAdminLogEventActionParticipantLeave`

**ID**

0xf89777f2

No parameters required.

#### 4.37.44 ChannelAdminLogEventActionParticipantToggleAdmin

`class pyrogram.api.types.ChannelAdminLogEventActionParticipantToggleAdmin`

**ID**

0xd5676710

**Parameters**

- **prev\_participant** – Either *ChannelParticipant*, *ChannelParticipantSelf*, *ChannelParticipantCreator*, *ChannelParticipantAdmin* or *ChannelParticipantBanned*
- **new\_participant** – Either *ChannelParticipant*, *ChannelParticipantSelf*, *ChannelParticipantCreator*, *ChannelParticipantAdmin* or *ChannelParticipantBanned*

#### 4.37.45 ChannelAdminLogEventActionParticipantToggleBan

`class pyrogram.api.types.ChannelAdminLogEventActionParticipantToggleBan`

**ID**

0xe6d83d7e

**Parameters**

- **prev\_participant** – Either *ChannelParticipant*, *ChannelParticipantSelf*, *ChannelParticipantCreator*, *ChannelParticipantAdmin* or *ChannelParticipantBanned*
- **new\_participant** – Either *ChannelParticipant*, *ChannelParticipantSelf*, *ChannelParticipantCreator*, *ChannelParticipantAdmin* or *ChannelParticipantBanned*

#### 4.37.46 ChannelAdminLogEventActionStopPoll

`class pyrogram.api.types.ChannelAdminLogEventActionStopPoll`

**ID**

0x8f079643

**Parameters** **message** – Either *MessageEmpty*, *Message* or *MessageService*

#### 4.37.47 ChannelAdminLogEventActionToggleInvites

```
class pyrogram.api.types.ChannelAdminLogEventActionToggleInvites
```

**ID**  
0x1b7907ae

**Parameters** `new_value` – bool

#### 4.37.48 ChannelAdminLogEventActionTogglePreHistoryHidden

```
class pyrogram.api.types.ChannelAdminLogEventActionTogglePreHistoryHidden
```

**ID**  
0x5f5c95f1

**Parameters** `new_value` – bool

#### 4.37.49 ChannelAdminLogEventActionToggleSignatures

```
class pyrogram.api.types.ChannelAdminLogEventActionToggleSignatures
```

**ID**  
0x26ae0971

**Parameters** `new_value` – bool

#### 4.37.50 ChannelAdminLogEventActionUpdatePinned

```
class pyrogram.api.types.ChannelAdminLogEventActionUpdatePinned
```

**ID**  
0xe9e82c18

**Parameters** `message` – Either *MessageEmpty*, *Message* or *MessageService*

#### 4.37.51 ChannelAdminLogEventsFilter

```
class pyrogram.api.types.ChannelAdminLogEventsFilter
```

**ID**  
0xea107ae4

**Parameters**

- `join` (*optional*) – bool
- `leave` (*optional*) – bool

- **invite**(*optional*) – bool
- **ban**(*optional*) – bool
- **unban**(*optional*) – bool
- **kick**(*optional*) – bool
- **unkick**(*optional*) – bool
- **promote**(*optional*) – bool
- **demote**(*optional*) – bool
- **info**(*optional*) – bool
- **settings**(*optional*) – bool
- **pinned**(*optional*) – bool
- **edit**(*optional*) – bool
- **delete**(*optional*) – bool

#### 4.37.52 ChannelForbidden

**class** pyrogram.api.types.ChannelForbidden

**ID**

0x289da732

**Parameters**

- **id** – int 32-bit
- **access\_hash** – int 64-bit
- **title** – str
- **broadcast**(*optional*) – bool
- **megagroup**(*optional*) – bool
- **until\_date**(*optional*) – int 32-bit

#### 4.37.53 ChannelFull

**class** pyrogram.api.types.ChannelFull

**ID**

0x1c87a71a

**Parameters**

- **id** – int 32-bit
- **about** – str
- **read\_inbox\_max\_id** – int 32-bit
- **read\_outbox\_max\_id** – int 32-bit

- **unread\_count** – int 32-bit
- **chat\_photo** – Either *PhotoEmpty* or *Photo*
- **notify\_settings** – *PeerNotifySettings*
- **exported\_invite** – Either *ChatInviteEmpty* or *ChatInviteExported*
- **bot\_info** – List of *BotInfo*
- **can\_view\_participants** (*optional*) – bool
- **can\_set\_username** (*optional*) – bool
- **can\_set\_stickers** (*optional*) – bool
- **hidden\_prehistory** (*optional*) – bool
- **can\_view\_stats** (*optional*) – bool
- **participants\_count** (*optional*) – int 32-bit
- **admins\_count** (*optional*) – int 32-bit
- **kicked\_count** (*optional*) – int 32-bit
- **banned\_count** (*optional*) – int 32-bit
- **online\_count** (*optional*) – int 32-bit
- **migrated\_from\_chat\_id** (*optional*) – int 32-bit
- **migrated\_from\_max\_id** (*optional*) – int 32-bit
- **pinned\_msg\_id** (*optional*) – int 32-bit
- **stickerset** (*optional*) – *StickerSet*
- **available\_min\_id** (*optional*) – int 32-bit

#### 4.37.54 ChannelMessagesFilter

**class** pyrogram.api.types.ChannelMessagesFilter

**ID**

0xcd77d957

**Parameters**

- **ranges** – List of *MessageRange*
- **exclude\_new\_messages** (*optional*) – bool

#### 4.37.55 ChannelMessagesFilterEmpty

**class** pyrogram.api.types.ChannelMessagesFilterEmpty

**ID**

0x94d42ee7

No parameters required.

#### 4.37.56 ChannelParticipant

```
class pyrogram.api.types.ChannelParticipant
```

**ID**  
0x15ebac1d

##### Parameters

- **user\_id** – int 32-bit
- **date** – int 32-bit

#### 4.37.57 ChannelParticipantAdmin

```
class pyrogram.api.types.ChannelParticipantAdmin
```

**ID**  
0x5daa6e23

##### Parameters

- **user\_id** – int 32-bit
- **promoted\_by** – int 32-bit
- **date** – int 32-bit
- **admin\_rights** – *ChatAdminRights*
- **can\_edit** (*optional*) – bool
- **is\_self** (*optional*) – bool
- **inviter\_id** (*optional*) – int 32-bit

#### 4.37.58 ChannelParticipantBanned

```
class pyrogram.api.types.ChannelParticipantBanned
```

**ID**  
0x1c0facaf

##### Parameters

- **user\_id** – int 32-bit
- **kicked\_by** – int 32-bit
- **date** – int 32-bit
- **banned\_rights** – *ChatBannedRights*
- **left** (*optional*) – bool

### 4.37.59 ChannelParticipantCreator

```
class pyrogram.api.types.ChannelParticipantCreator
```

**ID**

0xe3e2e1f9

**Parameters** `user_id` – int 32-bit

### 4.37.60 ChannelParticipantSelf

```
class pyrogram.api.types.ChannelParticipantSelf
```

**ID**

0xa3289a6d

**Parameters**

- `user_id` – int 32-bit
- `inviter_id` – int 32-bit
- `date` – int 32-bit

### 4.37.61 ChannelParticipantsAdmins

```
class pyrogram.api.types.ChannelParticipantsAdmins
```

**ID**

0xb4608969

No parameters required.

### 4.37.62 ChannelParticipantsBanned

```
class pyrogram.api.types.ChannelParticipantsBanned
```

**ID**

0x1427a5e1

**Parameters** `q` – str

### 4.37.63 ChannelParticipantsBots

```
class pyrogram.api.types.ChannelParticipantsBots
```

**ID**

0xb0d1865b

No parameters required.



#### 4.37.64 ChannelParticipantsContacts

```
class pyrogram.api.types.ChannelParticipantsContacts
```

**ID**  
0xbb6ae88d

**Parameters** *q* – str

#### 4.37.65 ChannelParticipantsKicked

```
class pyrogram.api.types.ChannelParticipantsKicked
```

**ID**  
0xa3b54985

**Parameters** *q* – str

#### 4.37.66 ChannelParticipantsRecent

```
class pyrogram.api.types.ChannelParticipantsRecent
```

**ID**  
0xde3f3c79

No parameters required.

#### 4.37.67 ChannelParticipantsSearch

```
class pyrogram.api.types.ChannelParticipantsSearch
```

**ID**  
0x0656ac4b

**Parameters** *q* – str

#### 4.37.68 Chat

```
class pyrogram.api.types.Chat
```

**ID**  
0x3bda1bde

**Parameters**

- **id** – int 32-bit
- **title** – str
- **photo** – Either *ChatPhotoEmpty* or *ChatPhoto*

- **participants\_count** – int 32-bit
- **date** – int 32-bit
- **version** – int 32-bit
- **creator** (*optional*) – bool
- **kicked** (*optional*) – bool
- **left** (*optional*) – bool
- **deactivated** (*optional*) – bool
- **migrated\_to** (*optional*) – Either *InputChannelEmpty* or *InputChannel*
- **admin\_rights** (*optional*) – *ChatAdminRights*
- **default\_banned\_rights** (*optional*) – *ChatBannedRights*

### 4.37.69 ChatAdminRights

**class** pyrogram.api.types.ChatAdminRights

**ID**

0x5fb224d5

**Parameters**

- **change\_info** (*optional*) – bool
- **post\_messages** (*optional*) – bool
- **edit\_messages** (*optional*) – bool
- **delete\_messages** (*optional*) – bool
- **ban\_users** (*optional*) – bool
- **invite\_users** (*optional*) – bool
- **pin\_messages** (*optional*) – bool
- **add\_admins** (*optional*) – bool

### 4.37.70 ChatBannedRights

**class** pyrogram.api.types.ChatBannedRights

**ID**

0x9f120418

**Parameters**

- **until\_date** – int 32-bit
- **view\_messages** (*optional*) – bool
- **send\_messages** (*optional*) – bool
- **send\_media** (*optional*) – bool

- **send\_stickers** (*optional*) – bool
- **send\_gifs** (*optional*) – bool
- **send\_games** (*optional*) – bool
- **send\_inline** (*optional*) – bool
- **embed\_links** (*optional*) – bool
- **send\_polls** (*optional*) – bool
- **change\_info** (*optional*) – bool
- **invite\_users** (*optional*) – bool
- **pin\_messages** (*optional*) – bool

#### 4.37.71 ChatEmpty

**class** pyrogram.api.types.ChatEmpty

**ID**

0x9ba2d800

**Parameters** **id** – int 32-bit

#### 4.37.72 ChatForbidden

**class** pyrogram.api.types.ChatForbidden

**ID**

0x07328bdb

**Parameters**

- **id** – int 32-bit
- **title** – str

#### 4.37.73 ChatFull

**class** pyrogram.api.types.ChatFull

**ID**

0x22a235da

**Parameters**

- **id** – int 32-bit
- **about** – str
- **participants** – Either *ChatParticipantsForbidden* or *ChatParticipants*
- **notify\_settings** – *PeerNotifySettings*

- **exported\_invite** – Either *ChatInviteEmpty* or *ChatInviteExported*
- **can\_set\_username** (*optional*) – bool
- **chat\_photo** (*optional*) – Either *PhotoEmpty* or *Photo*
- **bot\_info** (*optional*) – List of *BotInfo*
- **pinned\_msg\_id** (*optional*) – int 32-bit

#### 4.37.74 ChatInvite

**class** pyrogram.api.types.ChatInvite

**ID**

0xdb74f558

**Parameters**

- **title** – str
- **photo** – Either *ChatPhotoEmpty* or *ChatPhoto*
- **participants\_count** – int 32-bit
- **channel** (*optional*) – bool
- **broadcast** (*optional*) – bool
- **public** (*optional*) – bool
- **megagroup** (*optional*) – bool
- **participants** (*optional*) – List of either *UserEmpty* or *User*

**See also:**

This object can be returned by *messages.CheckChatInvite*.

#### 4.37.75 ChatInviteAlready

**class** pyrogram.api.types.ChatInviteAlready

**ID**

0x5a686d7c

**Parameters** **chat** – Either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*

**See also:**

This object can be returned by *messages.CheckChatInvite*.

#### 4.37.76 ChatInviteEmpty

**class** pyrogram.api.types.ChatInviteEmpty

**ID**

0x69df3769

No parameters required.

**See also:**

This object can be returned by *messages.ExportChatInvite* and *channels.ExportInvite*.

#### 4.37.77 ChatInviteExported

**class** pyrogram.api.types.ChatInviteExported

**ID**

0xfc2e05bc

**Parameters** *link* – str

**See also:**

This object can be returned by *messages.ExportChatInvite* and *channels.ExportInvite*.

#### 4.37.78 ChatOnline

**class** pyrogram.api.types.ChatOnline

**ID**

0xf041e250

**Parameters** *online* – int 32-bit

**See also:**

This object can be returned by *messages.GetOnline*.

#### 4.37.79 ChatParticipant

**class** pyrogram.api.types.ChatParticipant

**ID**

0xc8d7493e

**Parameters**

- **user\_id** – int 32-bit
- **inviter\_id** – int 32-bit
- **date** – int 32-bit

#### 4.37.80 ChatParticipantAdmin

```
class pyrogram.api.types.ChatParticipantAdmin
```

**ID**

0xe2d6e436

**Parameters**

- **user\_id** – int 32-bit
- **inviter\_id** – int 32-bit
- **date** – int 32-bit

#### 4.37.81 ChatParticipantCreator

```
class pyrogram.api.types.ChatParticipantCreator
```

**ID**

0xda13538a

**Parameters** **user\_id** – int 32-bit

#### 4.37.82 ChatParticipants

```
class pyrogram.api.types.ChatParticipants
```

**ID**

0x3f460fed

**Parameters**

- **chat\_id** – int 32-bit
- **participants** – List of either *ChatParticipant*, *ChatParticipantCreator* or *ChatParticipantAdmin*
- **version** – int 32-bit

#### 4.37.83 ChatParticipantsForbidden

```
class pyrogram.api.types.ChatParticipantsForbidden
```

**ID**

0xfc900c2b

**Parameters**

- **chat\_id** – int 32-bit
- **self\_participant** (optional) – Either *ChatParticipant*, *ChatParticipantCreator* or *ChatParticipantAdmin*

#### 4.37.84 ChatPhoto

**class** pyrogram.api.types.ChatPhoto

**ID**  
0x6153276a

**Parameters**

- **photo\_small** – Either *FileLocationUnavailable* or *FileLocation*
- **photo\_big** – Either *FileLocationUnavailable* or *FileLocation*

#### 4.37.85 ChatPhotoEmpty

**class** pyrogram.api.types.ChatPhotoEmpty

**ID**  
0x37c1011c

No parameters required.

#### 4.37.86 ClientDHIInnerData

**class** pyrogram.api.types.ClientDHIInnerData

**ID**  
0x6643b654

**Parameters**

- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **retry\_id** – int 64-bit
- **g\_b** – bytes

#### 4.37.87 CodeSettings

**class** pyrogram.api.types.CodeSettings

**ID**  
0x302f59f3

**Parameters**

- **allow\_flashcall** (*optional*) – bool
- **current\_number** (*optional*) – bool
- **app\_hash\_persistent** (*optional*) – bool
- **app\_hash** (*optional*) – str

### 4.37.88 Config

`class pyrogram.api.types.Config`

ID

0xe6ca25f6

#### Parameters

- **date** – int 32-bit
- **expires** – int 32-bit
- **test\_mode** – bool
- **this\_dc** – int 32-bit
- **dc\_options** – List of *DcOption*
- **dc\_txt\_domain\_name** – str
- **chat\_size\_max** – int 32-bit
- **megagroup\_size\_max** – int 32-bit
- **forwarded\_count\_max** – int 32-bit
- **online\_update\_period\_ms** – int 32-bit
- **offline\_blur\_timeout\_ms** – int 32-bit
- **offline\_idle\_timeout\_ms** – int 32-bit
- **online\_cloud\_timeout\_ms** – int 32-bit
- **notify\_cloud\_delay\_ms** – int 32-bit
- **notify\_default\_delay\_ms** – int 32-bit
- **push\_chat\_period\_ms** – int 32-bit
- **push\_chat\_limit** – int 32-bit
- **saved\_gifs\_limit** – int 32-bit
- **edit\_time\_limit** – int 32-bit
- **revoke\_time\_limit** – int 32-bit
- **revoke\_pm\_time\_limit** – int 32-bit
- **rating\_e\_decay** – int 32-bit
- **stickers\_recent\_limit** – int 32-bit
- **stickers\_faved\_limit** – int 32-bit
- **channels\_read\_media\_period** – int 32-bit
- **pinned\_dialogs\_count\_max** – int 32-bit
- **call\_receive\_timeout\_ms** – int 32-bit
- **call\_ring\_timeout\_ms** – int 32-bit
- **call\_connect\_timeout\_ms** – int 32-bit
- **call\_packet\_timeout\_ms** – int 32-bit



- **me\_url\_prefix** – str
- **caption\_length\_max** – int 32-bit
- **message\_length\_max** – int 32-bit
- **webfile\_dc\_id** – int 32-bit
- **phonecalls\_enabled** (*optional*) – bool
- **default\_p2p\_contacts** (*optional*) – bool
- **preload\_featured\_stickers** (*optional*) – bool
- **ignore\_phone\_entities** (*optional*) – bool
- **revoke\_pm\_inbox** (*optional*) – bool
- **blocked\_mode** (*optional*) – bool
- **pfs\_enabled** (*optional*) – bool
- **tmp\_sessions** (*optional*) – int 32-bit
- **autoupdate\_url\_prefix** (*optional*) – str
- **gif\_search\_username** (*optional*) – str
- **venue\_search\_username** (*optional*) – str
- **img\_search\_username** (*optional*) – str
- **static\_maps\_provider** (*optional*) – str
- **suggested\_lang\_code** (*optional*) – str
- **lang\_pack\_version** (*optional*) – int 32-bit
- **base\_lang\_pack\_version** (*optional*) – int 32-bit

See also:

This object can be returned by [help.GetConfig](#).

## 4.37.89 Contact

```
class pyrogram.api.types.Contact
```

**ID**

0xf911c994

**Parameters**

- **user\_id** – int 32-bit
- **mutual** – bool

## 4.37.90 ContactBlocked

```
class pyrogram.api.types.ContactBlocked
```

**ID**

0x561bc879

**Parameters**

- **user\_id** – int 32-bit
- **date** – int 32-bit

### 4.37.91 ContactLinkContact

```
class pyrogram.api.types.ContactLinkContact
```

**ID**

0xd502c2d0

No parameters required.

### 4.37.92 ContactLinkHasPhone

```
class pyrogram.api.types.ContactLinkHasPhone
```

**ID**

0x268f3f59

No parameters required.

### 4.37.93 ContactLinkNone

```
class pyrogram.api.types.ContactLinkNone
```

**ID**

0xfeedd3ad

No parameters required.

### 4.37.94 ContactLinkUnknown

```
class pyrogram.api.types.ContactLinkUnknown
```

**ID**

0x5f4f9247

No parameters required.

### 4.37.95 ContactStatus

```
class pyrogram.api.types.ContactStatus
```

**ID**

0xd3680c61

**Parameters**

- **user\_id** – int 32-bit
- **status** – Either *UserStatusEmpty*, *UserStatusOnline*, *UserStatusOffline*, *UserStatusRecently*, *UserStatusLastWeek* or *UserStatusLastMonth*

**See also:**

This object can be returned by *contacts.GetStatuses*.

### 4.37.96 DataJSON

```
class pyrogram.api.types.DataJSON
```

**ID**

0x7d748d04

**Parameters** **data** – str

**See also:**

This object can be returned by *bots.SendCustomRequest* and *phone.GetCallConfig*.

### 4.37.97 DcOption

```
class pyrogram.api.types.DcOption
```

**ID**

0x18b7a10d

**Parameters**

- **id** – int 32-bit
- **ip\_address** – str
- **port** – int 32-bit
- **ipv6** (*optional*) – bool
- **media\_only** (*optional*) – bool
- **tcpo\_only** (*optional*) – bool
- **cdn** (*optional*) – bool
- **static** (*optional*) – bool
- **secret** (*optional*) – bytes

#### 4.37.98 DestroyAuthKeyFail

**class** pyrogram.api.types.DestroyAuthKeyFail

**ID**

0xea109b13

No parameters required.

**See also:**

This object can be returned by [DestroyAuthKey](#).

#### 4.37.99 DestroyAuthKeyNone

**class** pyrogram.api.types.DestroyAuthKeyNone

**ID**

0x0a9f2259

No parameters required.

**See also:**

This object can be returned by [DestroyAuthKey](#).

#### 4.37.100 DestroyAuthKeyOk

**class** pyrogram.api.types.DestroyAuthKeyOk

**ID**

0xf660e1d4

No parameters required.

**See also:**

This object can be returned by [DestroyAuthKey](#).

#### 4.37.101 DestroySessionNone

**class** pyrogram.api.types.DestroySessionNone

**ID**

0x62d350c9

**Parameters** `session_id` – int 64-bit

**See also:**

This object can be returned by [DestroySession](#).

### 4.37.102 DestroySessionOk

```
class pyrogram.api.types.DestroySessionOk
```

**ID**

0xe22045fc

**Parameters** `session_id` – int 64-bit

**See also:**

This object can be returned by [\*DestroySession\*](#).

### 4.37.103 DhGenFail

```
class pyrogram.api.types.DhGenFail
```

**ID**

0xa69dae02

**Parameters**

- `nonce` – int 128-bit
- `server_nonce` – int 128-bit
- `new_nonce_hash3` – int 128-bit

**See also:**

This object can be returned by [\*SetClientDHParams\*](#).

### 4.37.104 DhGenOk

```
class pyrogram.api.types.DhGenOk
```

**ID**

0x3bcbf734

**Parameters**

- `nonce` – int 128-bit
- `server_nonce` – int 128-bit
- `new_nonce_hash1` – int 128-bit

**See also:**

This object can be returned by [\*SetClientDHParams\*](#).

### 4.37.105 DhGenRetry

**class** pyrogram.api.types.DhGenRetry

**ID**

0x46dc1fb9

**Parameters**

- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **new\_nonce\_hash2** – int 128-bit

**See also:**

This object can be returned by [\*SetClientDHPParams\*](#).

### 4.37.106 Dialog

**class** pyrogram.api.types.Dialog

**ID**

0xe4def5db

**Parameters**

- **peer** – Either [\*PeerUser\*](#), [\*PeerChat\*](#) or [\*PeerChannel\*](#)
- **top\_message** – int 32-bit
- **read\_inbox\_max\_id** – int 32-bit
- **read\_outbox\_max\_id** – int 32-bit
- **unread\_count** – int 32-bit
- **unread\_mentions\_count** – int 32-bit
- **notify\_settings** – [\*PeerNotifySettings\*](#)
- **pinned** (*optional*) – bool
- **unread\_mark** (*optional*) – bool
- **pts** (*optional*) – int 32-bit
- **draft** (*optional*) – Either [\*DraftMessageEmpty\*](#) or [\*DraftMessage\*](#)

### 4.37.107 DialogPeer

**class** pyrogram.api.types.DialogPeer

**ID**

0xe56dbf05

**Parameters** **peer** – Either [\*PeerUser\*](#), [\*PeerChat\*](#) or [\*PeerChannel\*](#)

**See also:**

This object can be returned by `messages.GetDialogUnreadMarks`.

**4.37.108 Document**

```
class pyrogram.api.types.Document
```

**ID**

0x9ba29cc1

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **file\_reference** – bytes
- **date** – int 32-bit
- **mime\_type** – str
- **size** – int 32-bit
- **dc\_id** – int 32-bit
- **attributes** – List of either `DocumentAttributeImageSize`, `DocumentAttributeAnimated`, `DocumentAttributeSticker`, `DocumentAttributeVideo`, `DocumentAttributeAudio`, `DocumentAttributeFilename` or `DocumentAttributeHasStickers`
- **thumbs** (optional) – List of either `PhotoSizeEmpty`, `PhotoSize`, `PhotoCachedSize` or `PhotoStrippedSize`

**See also:**

This object can be returned by `messages.GetDocumentByHash`.

**4.37.109 DocumentAttributeAnimated**

```
class pyrogram.api.types.DocumentAttributeAnimated
```

**ID**

0x11b58939

No parameters required.

**4.37.110 DocumentAttributeAudio**

```
class pyrogram.api.types.DocumentAttributeAudio
```

**ID**

0x9852f9c6

**Parameters**

- **duration** – int 32-bit
- **voice** (*optional*) – bool
- **title** (*optional*) – str
- **performer** (*optional*) – str
- **waveform** (*optional*) – bytes

#### 4.37.111 DocumentAttributeFilename

```
class pyrogram.api.types.DocumentAttributeFilename
```

**ID**

0x15590068

**Parameters** **file\_name** – str

#### 4.37.112 DocumentAttributeHasStickers

```
class pyrogram.api.types.DocumentAttributeHasStickers
```

**ID**

0x9801d2f7

No parameters required.

#### 4.37.113 DocumentAttributeImageSize

```
class pyrogram.api.types.DocumentAttributeImageSize
```

**ID**

0x6c37c15c

**Parameters**

- **w** – int 32-bit
- **h** – int 32-bit

#### 4.37.114 DocumentAttributeSticker

```
class pyrogram.api.types.DocumentAttributeSticker
```

**ID**

0x6319d612

**Parameters**

- **alt** – str



- **stickerset** – Either *InputStickerSetEmpty*, *InputStickerSetID* or *InputStickerSetShortName*
- **mask** (*optional*) – bool
- **mask\_coords** (*optional*) – *MaskCoords*

#### 4.37.115 DocumentAttributeVideo

**class** pyrogram.api.types.DocumentAttributeVideo

**ID**

0x0ef02ce6

**Parameters**

- **duration** – int 32-bit
- **w** – int 32-bit
- **h** – int 32-bit
- **round\_message** (*optional*) – bool
- **supports\_streaming** (*optional*) – bool

#### 4.37.116 DocumentEmpty

**class** pyrogram.api.types.DocumentEmpty

**ID**

0x36f8c871

**Parameters** **id** – int 64-bit

**See also:**

This object can be returned by *messages.GetDocumentByHash*.

#### 4.37.117 DraftMessage

**class** pyrogram.api.types.DraftMessage

**ID**

0xfd8e711f

**Parameters**

- **message** – str
- **date** – int 32-bit
- **no\_webpage** (*optional*) – bool
- **reply\_to\_msg\_id** (*optional*) – int 32-bit

- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

#### 4.37.118 DraftMessageEmpty

**class** pyrogram.api.types.DraftMessageEmpty

**ID**

0x1b0c841a

**Parameters** **date** (*optional*) – int 32-bit

#### 4.37.119 EmojiKeyword

**class** pyrogram.api.types.EmojiKeyword

**ID**

0xd5b3b9f9

**Parameters**

- **keyword** – str
- **emoticons** – List of str

#### 4.37.120 EmojiKeywordDeleted

**class** pyrogram.api.types.EmojiKeywordDeleted

**ID**

0x236df622

**Parameters**

- **keyword** – str
- **emoticons** – List of str

#### 4.37.121 EmojiKeywordsDifference

**class** pyrogram.api.types.EmojiKeywordsDifference

**ID**

0x5cc761bd

**Parameters**

- **lang\_code** – str
- **from\_version** – int 32-bit
- **version** – int 32-bit
- **keywords** – List of either *EmojiKeyword* or *EmojiKeywordDeleted*

**See also:**

This object can be returned by *messages.GetEmojiKeywords* and *messages.GetEmojiKeywordsDifference*.

**4.37.122 EmojiURL**

**class** pyrogram.api.types.**EmojiURL**

**ID**

0xa575739d

**Parameters** **url** – str

**See also:**

This object can be returned by *messages.GetEmojiURL*.

**4.37.123 EncryptedChat**

**class** pyrogram.api.types.**EncryptedChat**

**ID**

0xfa56ce36

**Parameters**

- **id** – int 32-bit
- **access\_hash** – int 64-bit
- **date** – int 32-bit
- **admin\_id** – int 32-bit
- **participant\_id** – int 32-bit
- **g\_a\_or\_b** – bytes
- **key\_fingerprint** – int 64-bit

**See also:**

This object can be returned by *messages.RequestEncryption* and *messages.AcceptEncryption*.

#### 4.37.124 EncryptedChatDiscarded

**class** pyrogram.api.types.**EncryptedChatDiscarded**

**ID**  
0x13d6dd27

**Parameters** **id** – int 32-bit

**See also:**

This object can be returned by *messages.RequestEncryption* and *messages.AcceptEncryption*.

#### 4.37.125 EncryptedChatEmpty

**class** pyrogram.api.types.**EncryptedChatEmpty**

**ID**  
0xab7ec0a0

**Parameters** **id** – int 32-bit

**See also:**

This object can be returned by *messages.RequestEncryption* and *messages.AcceptEncryption*.

#### 4.37.126 EncryptedChatRequested

**class** pyrogram.api.types.**EncryptedChatRequested**

**ID**  
0xc878527e

**Parameters**

- **id** – int 32-bit
- **access\_hash** – int 64-bit
- **date** – int 32-bit
- **admin\_id** – int 32-bit
- **participant\_id** – int 32-bit
- **g\_a** – bytes

**See also:**

This object can be returned by *messages.RequestEncryption* and *messages.AcceptEncryption*.

### 4.37.127 EncryptedChatWaiting

**class** pyrogram.api.types.**EncryptedChatWaiting**

**ID**

0x3bf703dc

#### Parameters

- **id** – int 32-bit
- **access\_hash** – int 64-bit
- **date** – int 32-bit
- **admin\_id** – int 32-bit
- **participant\_id** – int 32-bit

**See also:**

This object can be returned by *messages.RequestEncryption* and *messages.AcceptEncryption*.

### 4.37.128 EncryptedFile

**class** pyrogram.api.types.**EncryptedFile**

**ID**

0x4a70994c

#### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **size** – int 32-bit
- **dc\_id** – int 32-bit
- **key\_fingerprint** – int 32-bit

**See also:**

This object can be returned by *messages.UploadEncryptedFile*.

### 4.37.129 EncryptedFileEmpty

**class** pyrogram.api.types.**EncryptedFileEmpty**

**ID**

0xc21f497e

No parameters required.

**See also:**

This object can be returned by `messages.UploadEncryptedFile`.

### 4.37.130 EncryptedMessage

**class** pyrogram.api.types.EncryptedMessage

**ID**

0xed18c118

**Parameters**

- **random\_id** – int 64-bit
- **chat\_id** – int 32-bit
- **date** – int 32-bit
- **bytes** – bytes
- **file** – Either *EncryptedFileEmpty* or *EncryptedFile*

### 4.37.131 EncryptedMessageService

**class** pyrogram.api.types.EncryptedMessageService

**ID**

0x23734b06

**Parameters**

- **random\_id** – int 64-bit
- **chat\_id** – int 32-bit
- **date** – int 32-bit
- **bytes** – bytes

### 4.37.132 ExportedMessageLink

**class** pyrogram.api.types.ExportedMessageLink

**ID**

0x5dab1af4

**Parameters**

- **link** – str
- **html** – str

**See also:**

This object can be returned by `channels.ExportMessageLink`.

**4.37.133 FileHash**

```
class pyrogram.api.types.FileHash
```

**ID**

0x6242c773

**Parameters**

- **offset** – int 32-bit
- **limit** – int 32-bit
- **hash** – bytes

**See also:**

This object can be returned by `upload.ReuploadCdnFile`, `upload.GetCdnFileHashes` and `upload.GetFilesHashes`.

**4.37.134 FileLocation**

```
class pyrogram.api.types.FileLocation
```

**ID**

0x091d11eb

**Parameters**

- **dc\_id** – int 32-bit
- **volume\_id** – int 64-bit
- **local\_id** – int 32-bit
- **secret** – int 64-bit
- **file\_reference** – bytes

**4.37.135 FileLocationUnavailable**

```
class pyrogram.api.types.FileLocationUnavailable
```

**ID**

0x7c596b46

**Parameters**

- **volume\_id** – int 64-bit
- **local\_id** – int 32-bit

- **secret** – int 64-bit

#### 4.37.136 FoundGif

**class** pyrogram.api.types.**FoundGif**

**ID**

0x162ecc1f

##### Parameters

- **url** – str
- **thumb\_url** – str
- **content\_url** – str
- **content\_type** – str
- **w** – int 32-bit
- **h** – int 32-bit

#### 4.37.137 FoundGifCached

**class** pyrogram.api.types.**FoundGifCached**

**ID**

0x9c750409

##### Parameters

- **url** – str
- **photo** – Either *PhotoEmpty* or *Photo*
- **document** – Either *DocumentEmpty* or *Document*

#### 4.37.138 Game

**class** pyrogram.api.types.**Game**

**ID**

0xbdf9653b

##### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **short\_name** – str
- **title** – str
- **description** – str



- **photo** – Either *PhotoEmpty* or *Photo*
- **document** (*optional*) – Either *DocumentEmpty* or *Document*

#### 4.37.139 GeoPoint

**class** pyrogram.api.types.**GeoPoint**

**ID**

0x0296f104

**Parameters**

- **long** – float 64-bit
- **lat** – float 64-bit
- **access\_hash** – int 64-bit

#### 4.37.140 GeoPointEmpty

**class** pyrogram.api.types.**GeoPointEmpty**

**ID**

0x1117dd5f

No parameters required.

#### 4.37.141 HighScore

**class** pyrogram.api.types.**HighScore**

**ID**

0x58fffc0

**Parameters**

- **pos** – int 32-bit
- **user\_id** – int 32-bit
- **score** – int 32-bit

#### 4.37.142 HttpWait

**class** pyrogram.api.types.**HttpWait**

**ID**

0x9299359f

**Parameters**

- **max\_delay** – int 32-bit

- **wait\_after** – int 32-bit
- **max\_wait** – int 32-bit

#### 4.37.143 ImportedContact

**class** pyrogram.api.types.ImportedContact

**ID**  
0xd0028438

**Parameters**

- **user\_id** – int 32-bit
- **client\_id** – int 64-bit

#### 4.37.144 InlineBotSwitchPM

**class** pyrogram.api.types.InlineBotSwitchPM

**ID**  
0x3c20629f

**Parameters**

- **text** – str
- **start\_param** – str

#### 4.37.145 InputAppEvent

**class** pyrogram.api.types.InputAppEvent

**ID**  
0x1d1b1245

**Parameters**

- **time** – float 64-bit
- **type** – str
- **peer** – int 64-bit
- **data** – Either *JsonNull*, *JsonBool*, *JsonNumber*, *JsonString*, *JsonArray* or *JsonObject*

#### 4.37.146 InputBotInlineMessageGame

**class** pyrogram.api.types.InputBotInlineMessageGame

ID

0x4b425864

**Parameters** `reply_markup` (optional) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

#### 4.37.147 InputBotInlineMessageID

```
class pyrogram.api.types.InputBotInlineMessageID
```

ID

0x890c3d89

**Parameters**

- **dc\_id** – int 32-bit
- **id** – int 64-bit
- **access\_hash** – int 64-bit

#### 4.37.148 InputBotInlineMessageMediaAuto

```
class pyrogram.api.types.InputBotInlineMessageMediaAuto
```

ID

0x3380c786

**Parameters**

- **message** – str
- **entities** (optional) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*
- **reply\_markup** (optional) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

#### 4.37.149 InputBotInlineMessageMediaContact

```
class pyrogram.api.types.InputBotInlineMessageMediaContact
```

ID

0xa6edbffd

**Parameters**

- **phone\_number** – str
- **first\_name** – str
- **last\_name** – str
- **vcard** – str
- **reply\_markup** (optional) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

#### 4.37.150 InputBotInlineMessageMediaGeo

**class** pyrogram.api.types.InputBotInlineMessageMediaGeo

**ID**

0xc1b15d65

##### Parameters

- **geo\_point** – Either *InputGeoPointEmpty* or *InputGeoPoint*
- **period** – int 32-bit
- **reply\_markup** (optional) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

#### 4.37.151 InputBotInlineMessageMediaVenue

**class** pyrogram.api.types.InputBotInlineMessageMediaVenue

**ID**

0x417bbf11

##### Parameters

- **geo\_point** – Either *InputGeoPointEmpty* or *InputGeoPoint*
- **title** – str
- **address** – str
- **provider** – str
- **venue\_id** – str
- **venue\_type** – str
- **reply\_markup** (optional) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

### 4.37.152 InputBotInlineMessageText

```
class pyrogram.api.types.InputBotInlineMessageText
```

ID

0x3dcd7a87

#### Parameters

- **message** – str
- **no\_webpage** (*optional*) – bool
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*
- **reply\_markup** (*optional*) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*

### 4.37.153 InputBotInlineResult

```
class pyrogram.api.types.InputBotInlineResult
```

ID

0x88bf9319

#### Parameters

- **id** – str
- **type** – str
- **send\_message** – Either *InputBotInlineMessageMediaAuto*, *InputBotInlineMessageText*, *InputBotInlineMessageMediaGeo*, *InputBotInlineMessageMediaVenue*, *InputBotInlineMessageMediaContact* or *InputBotInlineMessageGame*
- **title** (*optional*) – str
- **description** (*optional*) – str
- **url** (*optional*) – str
- **thumb** (*optional*) – *InputWebDocument*
- **content** (*optional*) – *InputWebDocument*

### 4.37.154 InputBotInlineResultDocument

```
class pyrogram.api.types.InputBotInlineResultDocument
```

**ID**

0xffff8fdc4

**Parameters**

- **id** – str
- **type** – str
- **document** – Either *InputDocumentEmpty* or *InputDocument*
- **send\_message** – Either *InputBotInlineMessageMediaAuto*, *InputBotInlineMessageText*, *InputBotInlineMessageMediaGeo*, *InputBotInlineMessageMediaVenue*, *InputBotInlineMessageMediaContact* or *InputBotInlineMessageGame*
- **title** (*optional*) – str
- **description** (*optional*) – str

### 4.37.155 InputBotInlineResultGame

**class** pyrogram.api.types.**InputBotInlineResultGame****ID**

0x4fa417f2

**Parameters**

- **id** – str
- **short\_name** – str
- **send\_message** – Either *InputBotInlineMessageMediaAuto*, *InputBotInlineMessageText*, *InputBotInlineMessageMediaGeo*, *InputBotInlineMessageMediaVenue*, *InputBotInlineMessageMediaContact* or *InputBotInlineMessageGame*

### 4.37.156 InputBotInlineResultPhoto

**class** pyrogram.api.types.**InputBotInlineResultPhoto****ID**

0xa8d864a7

**Parameters**

- **id** – str
- **type** – str
- **photo** – Either *InputPhotoEmpty* or *InputPhoto*
- **send\_message** – Either *InputBotInlineMessageMediaAuto*, *InputBotInlineMessageText*, *InputBotInlineMessageMediaGeo*, *InputBotInlineMessageMediaVenue*, *InputBotInlineMessageMediaContact* or *InputBotInlineMessageGame*

### 4.37.157 InputChannel

**class** pyrogram.api.types.InputChannel

**ID**  
0xafeb712e

**Parameters**

- **channel\_id** – int 32-bit
- **access\_hash** – int 64-bit

### 4.37.158 InputChannelEmpty

**class** pyrogram.api.types.InputChannelEmpty

**ID**  
0xee8c1e86

No parameters required.

### 4.37.159 InputChatPhoto

**class** pyrogram.api.types.InputChatPhoto

**ID**  
0x8953ad37

**Parameters** **id** – Either *InputPhotoEmpty* or *InputPhoto*

### 4.37.160 InputChatPhotoEmpty

**class** pyrogram.api.types.InputChatPhotoEmpty

**ID**  
0x1ca48f57

No parameters required.

### 4.37.161 InputChatUploadedPhoto

**class** pyrogram.api.types.InputChatUploadedPhoto

**ID**  
0x927c55b4

**Parameters** **file** – Either *InputFile* or *InputFileBig*

#### 4.37.162 InputCheckPasswordEmpty

```
class pyrogram.api.types.InputCheckPasswordEmpty
```

**ID**

0x9880f658

No parameters required.

#### 4.37.163 InputCheckPasswordSRP

```
class pyrogram.api.types.InputCheckPasswordSRP
```

**ID**

0xd27ff082

**Parameters**

- **srp\_id** – int 64-bit
- **A** – bytes
- **M1** – bytes

#### 4.37.164 InputClientProxy

```
class pyrogram.api.types.InputClientProxy
```

**ID**

0x75588b3f

**Parameters**

- **address** – str
- **port** – int 32-bit

#### 4.37.165 InputDialogPeer

```
class pyrogram.api.types.InputDialogPeer
```

**ID**

0xfcaafeb7

**Parameters** **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*



### 4.37.166 InputDocument

**class** pyrogram.api.types.InputDocument

**ID**  
0x1abfb575

#### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **file\_reference** – bytes

### 4.37.167 InputDocumentEmpty

**class** pyrogram.api.types.InputDocumentEmpty

**ID**  
0x72f0eaae

No parameters required.

### 4.37.168 InputDocumentFileLocation

**class** pyrogram.api.types.InputDocumentFileLocation

**ID**  
0x196683d9

#### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **file\_reference** – bytes

### 4.37.169 InputEncryptedChat

**class** pyrogram.api.types.InputEncryptedChat

**ID**  
0xf141b5e1

#### Parameters

- **chat\_id** – int 32-bit
- **access\_hash** – int 64-bit

#### 4.37.170 InputEncryptedFile

**class** pyrogram.api.types.InputEncryptedFile

**ID**  
0x5a17b5e5

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit

#### 4.37.171 InputEncryptedFileBigUploaded

**class** pyrogram.api.types.InputEncryptedFileBigUploaded

**ID**  
0x2dc173c8

**Parameters**

- **id** – int 64-bit
- **parts** – int 32-bit
- **key\_fingerprint** – int 32-bit

#### 4.37.172 InputEncryptedFileEmpty

**class** pyrogram.api.types.InputEncryptedFileEmpty

**ID**  
0x1837c364

No parameters required.

#### 4.37.173 InputEncryptedFileLocation

**class** pyrogram.api.types.InputEncryptedFileLocation

**ID**  
0xf5235d55

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit

#### 4.37.174 InputEncryptedFileUploaded

**class** pyrogram.api.types.InputEncryptedFileUploaded

**ID**

0x64bd0306

**Parameters**

- **id** – int 64-bit
- **parts** – int 32-bit
- **md5\_checksum** – str
- **key\_fingerprint** – int 32-bit

#### 4.37.175 InputFile

**class** pyrogram.api.types.InputFile

**ID**

0xf52ff27f

**Parameters**

- **id** – int 64-bit
- **parts** – int 32-bit
- **name** – str
- **md5\_checksum** – str

#### 4.37.176 InputFileBig

**class** pyrogram.api.types.InputFileBig

**ID**

0xfa4f0bb5

**Parameters**

- **id** – int 64-bit
- **parts** – int 32-bit
- **name** – str

#### 4.37.177 InputFileLocation

**class** pyrogram.api.types.InputFileLocation

**ID**

0xdfdaabel

**Parameters**

- **volume\_id** – int 64-bit
- **local\_id** – int 32-bit
- **secret** – int 64-bit
- **file\_reference** – bytes

**4.37.178 InputGameID**

```
class pyrogram.api.types.InputGameID
```

**ID**

```
0x032c3e77
```

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit

**4.37.179 InputGameShortName**

```
class pyrogram.api.types.InputGameShortName
```

**ID**

```
0xc331e80a
```

**Parameters**

- **bot\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*
- **short\_name** – str

**4.37.180 InputGeoPoint**

```
class pyrogram.api.types.InputGeoPoint
```

**ID**

```
0xf3b7acc9
```

**Parameters**

- **lat** – float 64-bit
- **long** – float 64-bit

### 4.37.181 InputGeoPointEmpty

```
class pyrogram.api.types.InputGeoPointEmpty
```

**ID**

0xe4c123d6

No parameters required.

### 4.37.182 InputMediaContact

```
class pyrogram.api.types.InputMediaContact
```

**ID**

0xf8ab7dfb

**Parameters**

- **phone\_number** – str
- **first\_name** – str
- **last\_name** – str
- **vcard** – str

### 4.37.183 InputMediaDocument

```
class pyrogram.api.types.InputMediaDocument
```

**ID**

0x23ab23d2

**Parameters**

- **id** – Either *InputDocumentEmpty* or *InputDocument*
- **ttl\_seconds** (*optional*) – int 32-bit

### 4.37.184 InputMediaDocumentExternal

```
class pyrogram.api.types.InputMediaDocumentExternal
```

**ID**

0xfb52dc99

**Parameters**

- **url** – str
- **ttl\_seconds** (*optional*) – int 32-bit

### 4.37.185 InputMediaEmpty

```
class pyrogram.api.types.InputMediaEmpty
```

**ID**

0x9664f57f

No parameters required.

### 4.37.186 InputMediaGame

```
class pyrogram.api.types.InputMediaGame
```

**ID**

0xd33f43f3

**Parameters** `id` – Either *InputGameID* or *InputGameShortName*

### 4.37.187 InputMediaGeoLive

```
class pyrogram.api.types.InputMediaGeoLive
```

**ID**

0xce4e82fd

**Parameters**

- **geo\_point** – Either *InputGeoPointEmpty* or *InputGeoPoint*
- **stopped**(*optional*) – bool
- **period**(*optional*) – int 32-bit

### 4.37.188 InputMediaGeoPoint

```
class pyrogram.api.types.InputMediaGeoPoint
```

**ID**

0xf9c44144

**Parameters** `geo_point` – Either *InputGeoPointEmpty* or *InputGeoPoint*

### 4.37.189 InputMediaGifExternal

```
class pyrogram.api.types.InputMediaGifExternal
```

**ID**

0x4843b0fd

**Parameters**

- **url** – str
- **q** – str

#### 4.37.190 InputMediaInvoice

**class** pyrogram.api.types.InputMediaInvoice

**ID**

0xf4e096c3

**Parameters**

- **title** – str
- **description** – str
- **invoice** – *Invoice*
- **payload** – bytes
- **provider** – str
- **provider\_data** – *DataJSON*
- **start\_param** – str
- **photo** (*optional*) – *InputWebDocument*

#### 4.37.191 InputMediaPhoto

**class** pyrogram.api.types.InputMediaPhoto

**ID**

0xb3ba0635

**Parameters**

- **id** – Either *InputPhotoEmpty* or *InputPhoto*
- **ttl\_seconds** (*optional*) – int 32-bit

#### 4.37.192 InputMediaPhotoExternal

**class** pyrogram.api.types.InputMediaPhotoExternal

**ID**

0xe5bbfela

**Parameters**

- **url** – str
- **ttl\_seconds** (*optional*) – int 32-bit

### 4.37.193 InputMediaPoll

```
class pyrogram.api.types.InputMediaPoll
```

ID

0x06b3765b

Parameters **poll** – *Poll*

### 4.37.194 InputMediaUploadedDocument

```
class pyrogram.api.types.InputMediaUploadedDocument
```

ID

0x5b38c6c1

Parameters

- **file** – Either *InputFile* or *InputFileBig*
- **mime\_type** – str
- **attributes** – List of either *DocumentAttributeImageSize*, *DocumentAttributeAnimated*, *DocumentAttributeSticker*, *DocumentAttributeVideo*, *DocumentAttributeAudio*, *DocumentAttributeFilename* or *DocumentAttributeHasStickers*
- **nosound\_video** (*optional*) – bool
- **thumb** (*optional*) – Either *InputFile* or *InputFileBig*
- **stickers** (*optional*) – List of either *InputDocumentEmpty* or *InputDocument*
- **ttl\_seconds** (*optional*) – int 32-bit

### 4.37.195 InputMediaUploadedPhoto

```
class pyrogram.api.types.InputMediaUploadedPhoto
```

ID

0x1e287d04

Parameters

- **file** – Either *InputFile* or *InputFileBig*
- **stickers** (*optional*) – List of either *InputDocumentEmpty* or *InputDocument*
- **ttl\_seconds** (*optional*) – int 32-bit

### 4.37.196 InputMediaVenue

```
class pyrogram.api.types.InputMediaVenue
```



**ID**

0xc13d1c11

**Parameters**

- **geo\_point** – Either *InputGeoPointEmpty* or *InputGeoPoint*
- **title** – str
- **address** – str
- **provider** – str
- **venue\_id** – str
- **venue\_type** – str

**4.37.197 InputMessageEntityMentionName****class** pyrogram.api.types.InputMessageEntityMentionName**ID**

0x208e68c9

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit
- **user\_id** – Either *InputUserEmpty*, *InputUserSelf* or *InputUser*

**4.37.198 InputMessageID****class** pyrogram.api.types.InputMessageID**ID**

0xa676a322

**Parameters** **id** – int 32-bit**4.37.199 InputMessagePinned****class** pyrogram.api.types.InputMessagePinned**ID**

0x86872538

No parameters required.

#### 4.37.200 InputMessageReplyTo

**class** pyrogram.api.types.InputMessageReplyTo

**ID**

0xbad88395

**Parameters** **id** – int 32-bit

#### 4.37.201 InputMessagesFilterChatPhotos

**class** pyrogram.api.types.InputMessagesFilterChatPhotos

**ID**

0x3a20ecb8

No parameters required.

#### 4.37.202 InputMessagesFilterContacts

**class** pyrogram.api.types.InputMessagesFilterContacts

**ID**

0xe062db83

No parameters required.

#### 4.37.203 InputMessagesFilterDocument

**class** pyrogram.api.types.InputMessagesFilterDocument

**ID**

0x9eddf188

No parameters required.

#### 4.37.204 InputMessagesFilterEmpty

**class** pyrogram.api.types.InputMessagesFilterEmpty

**ID**

0x57e2f66c

No parameters required.

#### 4.37.205 InputMessagesFilterGeo

```
class pyrogram.api.types.InputMessagesFilterGeo
```

**ID**

0xe7026d0d

No parameters required.

#### 4.37.206 InputMessagesFilterGif

```
class pyrogram.api.types.InputMessagesFilterGif
```

**ID**

0xffc86587

No parameters required.

#### 4.37.207 InputMessagesFilterMusic

```
class pyrogram.api.types.InputMessagesFilterMusic
```

**ID**

0x3751b49e

No parameters required.

#### 4.37.208 InputMessagesFilterMyMentions

```
class pyrogram.api.types.InputMessagesFilterMyMentions
```

**ID**

0xc1f8e69a

No parameters required.

#### 4.37.209 InputMessagesFilterPhoneCalls

```
class pyrogram.api.types.InputMessagesFilterPhoneCalls
```

**ID**

0x80c99768

**Parameters missed** (*optional*) – bool

#### 4.37.210 InputMessagesFilterPhotoVideo

```
class pyrogram.api.types.InputMessagesFilterPhotoVideo
```

**ID**

0x56e9f0e4

No parameters required.

#### 4.37.211 InputMessagesFilterPhotos

```
class pyrogram.api.types.InputMessagesFilterPhotos
```

**ID**

0x9609a51c

No parameters required.

#### 4.37.212 InputMessagesFilterRoundVideo

```
class pyrogram.api.types.InputMessagesFilterRoundVideo
```

**ID**

0xb549da53

No parameters required.

#### 4.37.213 InputMessagesFilterRoundVoice

```
class pyrogram.api.types.InputMessagesFilterRoundVoice
```

**ID**

0x7a7c17a4

No parameters required.

#### 4.37.214 InputMessagesFilterUrl

```
class pyrogram.api.types.InputMessagesFilterUrl
```

**ID**

0x7ef0dd87

No parameters required.

#### 4.37.215 InputMessagesFilterVideo

```
class pyrogram.api.types.InputMessagesFilterVideo
```

**ID**

0x9fc00e65

No parameters required.

#### 4.37.216 InputMessagesFilterVoice

```
class pyrogram.api.types.InputMessagesFilterVoice
```

**ID**

0x50f5c392

No parameters required.

#### 4.37.217 InputNotifyBroadcasts

```
class pyrogram.api.types.InputNotifyBroadcasts
```

**ID**

0xb1db7c7e

No parameters required.

#### 4.37.218 InputNotifyChats

```
class pyrogram.api.types.InputNotifyChats
```

**ID**

0x4a95e84e

No parameters required.

#### 4.37.219 InputNotifyPeer

```
class pyrogram.api.types.InputNotifyPeer
```

**ID**

0xb8bc5b0c

**Parameters** **peer** – Either *InputPeerEmpty*, *InputPeerSelf*, *InputPeerChat*, *InputPeerUser* or *InputPeerChannel*

#### 4.37.220 InputNotifyUsers

```
class pyrogram.api.types.InputNotifyUsers
```

**ID**

0x193b4417

No parameters required.

#### 4.37.221 InputPaymentCredentials

```
class pyrogram.api.types.InputPaymentCredentials
```

**ID**

0x3417d728

**Parameters**

- **data** – *DataJSON*
- **save** (*optional*) – bool

#### 4.37.222 InputPaymentCredentialsAndroidPay

```
class pyrogram.api.types.InputPaymentCredentialsAndroidPay
```

**ID**

0xca05d50e

**Parameters**

- **payment\_token** – *DataJSON*
- **google\_transaction\_id** – str

#### 4.37.223 InputPaymentCredentialsApplePay

```
class pyrogram.api.types.InputPaymentCredentialsApplePay
```

**ID**

0x0aa1c39f

**Parameters** **payment\_data** – *DataJSON*

#### 4.37.224 InputPaymentCredentialsSaved

```
class pyrogram.api.types.InputPaymentCredentialsSaved
```

**ID**

0xc10eb2cf

**Parameters**

- **id** – str
- **tmp\_password** – bytes

**4.37.225 InputPeerChannel**

```
class pyrogram.api.types.InputPeerChannel
```

**ID**

0x20adaef8

**Parameters**

- **channel\_id** – int 32-bit
- **access\_hash** – int 64-bit

**4.37.226 InputPeerChat**

```
class pyrogram.api.types.InputPeerChat
```

**ID**

0x179be863

**Parameters** **chat\_id** – int 32-bit

**4.37.227 InputPeerEmpty**

```
class pyrogram.api.types.InputPeerEmpty
```

**ID**

0x7f3b18ea

No parameters required.

**4.37.228 InputPeerNotifySettings**

```
class pyrogram.api.types.InputPeerNotifySettings
```

**ID**

0x9c3d198e

**Parameters**

- **show\_previews** (*optional*) – bool
- **silent** (*optional*) – bool
- **mute\_until** (*optional*) – int 32-bit
- **sound** (*optional*) – str

#### 4.37.229 InputPeerSelf

```
class pyrogram.api.types.InputPeerSelf
```

**ID**

0x7da07ec9

No parameters required.

#### 4.37.230 InputPeerUser

```
class pyrogram.api.types.InputPeerUser
```

**ID**

0x7b8e7de6

**Parameters**

- **user\_id** – int 32-bit
- **access\_hash** – int 64-bit

#### 4.37.231 InputPhoneCall

```
class pyrogram.api.types.InputPhoneCall
```

**ID**

0x1e36fded

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit

#### 4.37.232 InputPhoneContact

```
class pyrogram.api.types.InputPhoneContact
```

**ID**

0xf392b7f4

**Parameters**

- **client\_id** – int 64-bit
- **phone** – str
- **first\_name** – str
- **last\_name** – str



### 4.37.233 InputPhoto

```
class pyrogram.api.types.InputPhoto
```

**ID**  
0x3bb3b94a

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **file\_reference** – bytes

### 4.37.234 InputPhotoEmpty

```
class pyrogram.api.types.InputPhotoEmpty
```

**ID**  
0x1cd7bf0d

No parameters required.

### 4.37.235 InputPrivacyKeyChatInvite

```
class pyrogram.api.types.InputPrivacyKeyChatInvite
```

**ID**  
0xbdfb0426

No parameters required.

### 4.37.236 InputPrivacyKeyForwards

```
class pyrogram.api.types.InputPrivacyKeyForwards
```

**ID**  
0xa4dd4c08

No parameters required.

### 4.37.237 InputPrivacyKeyPhoneCall

```
class pyrogram.api.types.InputPrivacyKeyPhoneCall
```

**ID**  
0xfabadc5f

No parameters required.

#### 4.37.238 InputPrivacyKeyPhoneP2P

```
class pyrogram.api.types.InputPrivacyKeyPhoneP2P
```

**ID**

0xdb9e70d2

No parameters required.

#### 4.37.239 InputPrivacyKeyProfilePhoto

```
class pyrogram.api.types.InputPrivacyKeyProfilePhoto
```

**ID**

0x5719bacc

No parameters required.

#### 4.37.240 InputPrivacyKeyStatusTimestamp

```
class pyrogram.api.types.InputPrivacyKeyStatusTimestamp
```

**ID**

0x4f96cb18

No parameters required.

#### 4.37.241 InputPrivacyValueAllowAll

```
class pyrogram.api.types.InputPrivacyValueAllowAll
```

**ID**

0x184b35ce

No parameters required.

#### 4.37.242 InputPrivacyValueAllowContacts

```
class pyrogram.api.types.InputPrivacyValueAllowContacts
```

**ID**

0x0d09e07b

No parameters required.

#### 4.37.243 InputPrivacyValueAllowUsers

**class** pyrogram.api.types.InputPrivacyValueAllowUsers

**ID**

0x131cc67f

**Parameters** **users** – List of either *InputUserEmpty*, *InputUserSelf* or *InputUser*

#### 4.37.244 InputPrivacyValueDisallowAll

**class** pyrogram.api.types.InputPrivacyValueDisallowAll

**ID**

0xd66b66c9

No parameters required.

#### 4.37.245 InputPrivacyValueDisallowContacts

**class** pyrogram.api.types.InputPrivacyValueDisallowContacts

**ID**

0x0ba52007

No parameters required.

#### 4.37.246 InputPrivacyValueDisallowUsers

**class** pyrogram.api.types.InputPrivacyValueDisallowUsers

**ID**

0x90110467

**Parameters** **users** – List of either *InputUserEmpty*, *InputUserSelf* or *InputUser*

#### 4.37.247 InputReportReasonChildAbuse

**class** pyrogram.api.types.InputReportReasonChildAbuse

**ID**

0xadf44ee3

No parameters required.

#### 4.37.248 InputReportReasonCopyright

```
class pyrogram.api.types.InputReportReasonCopyright
```

**ID**

0x9b89f93a

No parameters required.

#### 4.37.249 InputReportReasonOther

```
class pyrogram.api.types.InputReportReasonOther
```

**ID**

0xe1746d0a

**Parameters** `text` – str

#### 4.37.250 InputReportReasonPornography

```
class pyrogram.api.types.InputReportReasonPornography
```

**ID**

0x2e59d922

No parameters required.

#### 4.37.251 InputReportReasonSpam

```
class pyrogram.api.types.InputReportReasonSpam
```

**ID**

0x58dbcab8

No parameters required.

#### 4.37.252 InputReportReasonViolence

```
class pyrogram.api.types.InputReportReasonViolence
```

**ID**

0x1e22c78d

No parameters required.

#### 4.37.253 InputSecureFile

**class** pyrogram.api.types.InputSecureFile

**ID**  
0x5367e5be

##### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit

#### 4.37.254 InputSecureFileLocation

**class** pyrogram.api.types.InputSecureFileLocation

**ID**  
0xcbc7ee28

##### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit

#### 4.37.255 InputSecureFileUploaded

**class** pyrogram.api.types.InputSecureFileUploaded

**ID**  
0x3334b0f0

##### Parameters

- **id** – int 64-bit
- **parts** – int 32-bit
- **md5\_checksum** – str
- **file\_hash** – bytes
- **secret** – bytes

#### 4.37.256 InputSecureValue

**class** pyrogram.api.types.InputSecureValue

**ID**  
0xdb21d0a7

##### Parameters

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **data** (optional) – *SecureData*
- **front\_side** (optional) – Either *InputSecureFileUploaded* or *InputSecureFile*
- **reverse\_side** (optional) – Either *InputSecureFileUploaded* or *InputSecureFile*
- **selfie** (optional) – Either *InputSecureFileUploaded* or *InputSecureFile*
- **translation** (optional) – List of either *InputSecureFileUploaded* or *InputSecureFile*
- **files** (optional) – List of either *InputSecureFileUploaded* or *InputSecureFile*
- **plain\_data** (optional) – Either *SecurePlainPhone* or *SecurePlainEmail*

### 4.37.257 InputSingleMedia

**class** pyrogram.api.types.InputSingleMedia

**ID**

0x1cc6e91f

#### Parameters

- **media** – Either *InputMediaEmpty*, *InputMediaUploadedPhoto*, *InputMediaPhoto*, *InputMediaGeoPoint*, *InputMediaContact*, *InputMediaUploadedDocument*, *InputMediaDocument*, *InputMediaVenue*, *InputMediaGifExternal*, *InputMediaPhotoExternal*, *InputMediaDocumentExternal*, *InputMediaGame*, *InputMediaInvoice*, *InputMediaGeoLive* or *InputMediaPoll*
- **random\_id** – int 64-bit
- **message** – str
- **entities** (optional) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

### 4.37.258 InputStickerSetEmpty

**class** pyrogram.api.types.InputStickerSetEmpty

**ID**

0xffb62b95

No parameters required.

### 4.37.259 InputStickerSetID

**class** pyrogram.api.types.InputStickerSetID

**ID**

0x9de7a269

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit

### 4.37.260 InputStickerSetItem

**class** pyrogram.api.types.InputStickerSetItem

**ID**

0xffa0a496

**Parameters**

- **document** – Either *InputDocumentEmpty* or *InputDocument*
- **emoji** – str
- **mask\_coords** (*optional*) – *MaskCoords*

### 4.37.261 InputStickerSetShortName

**class** pyrogram.api.types.InputStickerSetShortName

**ID**

0x861cc8a0

**Parameters** **short\_name** – str

### 4.37.262 InputStickeredMediaDocument

**class** pyrogram.api.types.InputStickeredMediaDocument

**ID**

0x0438865b

**Parameters** *id* – Either *InputDocumentEmpty* or *InputDocument*

#### 4.37.263 InputStickeredMediaPhoto

**class** pyrogram.api.types.InputStickeredMediaPhoto

**ID**

0x4a992157

**Parameters** *id* – Either *InputPhotoEmpty* or *InputPhoto*

#### 4.37.264 InputTakeoutFileLocation

**class** pyrogram.api.types.InputTakeoutFileLocation

**ID**

0x29be5899

No parameters required.

#### 4.37.265 InputUser

**class** pyrogram.api.types.InputUser

**ID**

0xd8292816

**Parameters**

- **user\_id** – int 32-bit
- **access\_hash** – int 64-bit

#### 4.37.266 InputUserEmpty

**class** pyrogram.api.types.InputUserEmpty

**ID**

0xb98886cf

No parameters required.



#### 4.37.267 InputUserSelf

**class** pyrogram.api.types.InputUserSelf

**ID**

0xf7c1b13f

No parameters required.

#### 4.37.268 InputWallPaper

**class** pyrogram.api.types.InputWallPaper

**ID**

0xe630b979

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit

#### 4.37.269 InputWallPaperSlug

**class** pyrogram.api.types.InputWallPaperSlug

**ID**

0x72091c80

**Parameters** **slug** – str

#### 4.37.270 InputWebDocument

**class** pyrogram.api.types.InputWebDocument

**ID**

0x9bed434d

**Parameters**

- **url** – str
- **size** – int 32-bit
- **mime\_type** – str
- **attributes** – List of either *DocumentAttributeImageSize*,  
*DocumentAttributeAnimated*, *DocumentAttributeSticker*,  
*DocumentAttributeVideo*, *DocumentAttributeAudio*,  
*DocumentAttributeFilename* or *DocumentAttributeHasStickers*

### 4.37.271 InputWebFileGeoPointLocation

```
class pyrogram.api.types.InputWebFileGeoPointLocation
```

ID

0x9f2221c9

#### Parameters

- **geo\_point** – Either *InputGeoPointEmpty* or *InputGeoPoint*
- **access\_hash** – int 64-bit
- **w** – int 32-bit
- **h** – int 32-bit
- **zoom** – int 32-bit
- **scale** – int 32-bit

### 4.37.272 InputWebFileLocation

```
class pyrogram.api.types.InputWebFileLocation
```

ID

0xc239d686

#### Parameters

- **url** – str
- **access\_hash** – int 64-bit

### 4.37.273 Invoice

```
class pyrogram.api.types.Invoice
```

ID

0xc30aa358

#### Parameters

- **currency** – str
- **prices** – List of *LabeledPrice*
- **test** (*optional*) – bool
- **name\_requested** (*optional*) – bool
- **phone\_requested** (*optional*) – bool
- **email\_requested** (*optional*) – bool
- **shipping\_address\_requested** (*optional*) – bool
- **flexible** (*optional*) – bool

- **phone\_to\_provider** (*optional*) – bool
- **email\_to\_provider** (*optional*) – bool

#### 4.37.274 IpPort

**class** pyrogram.api.types.**IpPort**

**ID**

0xd433ad73

**Parameters**

- **ipv4** – int 32-bit
- **port** – int 32-bit

#### 4.37.275 IpPortSecret

**class** pyrogram.api.types.**IpPortSecret**

**ID**

0x37982646

**Parameters**

- **ipv4** – int 32-bit
- **port** – int 32-bit
- **secret** – bytes

#### 4.37.276 JsonArray

**class** pyrogram.api.types.**JsonArray**

**ID**

0xf7444763

**Parameters value** – List of either *JsonNull*, *JsonBool*, *JsonNumber*, *JsonString*, *JsonArray* or *JsonObject*

**See also:**

This object can be returned by *help.GetAppConfig*.

#### 4.37.277 JsonBool

**class** pyrogram.api.types.**JsonBool**

**ID**

0xc7345e6a

**Parameters** **value** – bool

**See also:**

This object can be returned by [\*help.GetAppConfig\*](#).

#### 4.37.278 JsonNull

**class** pyrogram.api.types.JsonNull

**ID**

0x3f6d7b68

No parameters required.

**See also:**

This object can be returned by [\*help.GetAppConfig\*](#).

#### 4.37.279 JsonNumber

**class** pyrogram.api.types.JsonNumber

**ID**

0x2be0dfa4

**Parameters** **value** – float 64-bit

**See also:**

This object can be returned by [\*help.GetAppConfig\*](#).

#### 4.37.280 JsonObject

**class** pyrogram.api.types.JsonObject

**ID**

0x99c1d49d

**Parameters** **value** – List of [\*JsonObjectValue\*](#)

**See also:**

This object can be returned by [\*help.GetAppConfig\*](#).

#### 4.37.281 JsonObjectValue

**class** pyrogram.api.types.JsonObjectValue

**ID**

0xc0de1bd9

**Parameters**

- **key** – str
- **value** – Either *JsonNull*, *JsonBool*, *JsonNumber*, *JsonString*, *JsonArray* or *JsonObject*

**4.37.282 JsonString**

```
class pyrogram.api.types.JsonString
```

**ID**

0xb71e767a

**Parameters** **value** – str

**See also:**

This object can be returned by *help.GetAppConfig*.

**4.37.283 KeyboardButton**

```
class pyrogram.api.types.KeyboardButton
```

**ID**

0xa2fa4880

**Parameters** **text** – str

**4.37.284 KeyboardButtonBuy**

```
class pyrogram.api.types.KeyboardButtonBuy
```

**ID**

0xafd93fbb

**Parameters** **text** – str

**4.37.285 KeyboardButtonCallback**

```
class pyrogram.api.types.KeyboardButtonCallback
```

**ID**

0x683a5e46

**Parameters**

- **text** – str
- **data** – bytes

#### 4.37.286 KeyboardButtonGame

```
class pyrogram.api.types.KeyboardButtonGame
```

```
 ID
 0x50f41ccf
```

```
 Parameters text – str
```

#### 4.37.287 KeyboardButtonRequestGeoLocation

```
class pyrogram.api.types.KeyboardButtonRequestGeoLocation
```

```
 ID
 0xfc796b3f
```

```
 Parameters text – str
```

#### 4.37.288 KeyboardButtonRequestPhone

```
class pyrogram.api.types.KeyboardButtonRequestPhone
```

```
 ID
 0xb16a6c29
```

```
 Parameters text – str
```

#### 4.37.289 KeyboardButtonRow

```
class pyrogram.api.types.KeyboardButtonRow
```

```
 ID
 0x77608b83
```

```
 Parameters buttons – List of either KeyboardButton, KeyboardButtonUrl,
 KeyboardButtonCallback, KeyboardButtonRequestPhone,
 KeyboardButtonRequestGeoLocation, KeyboardButtonSwitchInline,
 KeyboardButtonGame or KeyboardButtonBuy
```

#### 4.37.290 KeyboardButtonSwitchInline

```
class pyrogram.api.types.KeyboardButtonSwitchInline
```

```
 ID
 0x0568a748
```

```
 Parameters
```

- **text** – str
- **query** – str
- **same\_peer** (*optional*) – bool

#### 4.37.291 KeyboardButtonUrl

**class** pyrogram.api.types.KeyboardButtonUrl

**ID**

0x258aff05

**Parameters**

- **text** – str
- **url** – str

#### 4.37.292 LabeledPrice

**class** pyrogram.api.types.LabeledPrice

**ID**

0xcb296bf8

**Parameters**

- **label** – str
- **amount** – int 64-bit

#### 4.37.293 LangPackDifference

**class** pyrogram.api.types.LangPackDifference

**ID**

0xf385c1f6

**Parameters**

- **lang\_code** – str
- **from\_version** – int 32-bit
- **version** – int 32-bit
- **strings** – List of either *LangPackString*, *LangPackStringPluralized* or *LangPackStringDeleted*

**See also:**

This object can be returned by *langpack.GetLangPack* and *langpack.GetDifference*.

### 4.37.294 LangPackLanguage

**class** pyrogram.api.types.LangPackLanguage

**ID**

0xeeca5ce3

**Parameters**

- **name** – str
- **native\_name** – str
- **lang\_code** – str
- **plural\_code** – str
- **strings\_count** – int 32-bit
- **translated\_count** – int 32-bit
- **translations\_url** – str
- **official** (*optional*) – bool
- **rtl** (*optional*) – bool
- **beta** (*optional*) – bool
- **base\_lang\_code** (*optional*) – str

**See also:**

This object can be returned by [\*langpack.GetLanguages\*](#) and [\*langpack.GetLanguage\*](#).

### 4.37.295 LangPackString

**class** pyrogram.api.types.LangPackString

**ID**

0xcad181f6

**Parameters**

- **key** – str
- **value** – str

**See also:**

This object can be returned by [\*langpack.GetStrings\*](#).

### 4.37.296 LangPackStringDeleted

**class** pyrogram.api.types.LangPackStringDeleted

**ID**

0x2979eeb2



**Parameters** `key` – str

**See also:**

This object can be returned by `langpack.GetStrings`.

### 4.37.297 LangPackStringPluralized

**class** `pyrogram.api.types.LangPackStringPluralized`

**ID**

0x6c47ac9f

**Parameters**

- `key` – str
- `other_value` – str
- `zero_value` (*optional*) – str
- `one_value` (*optional*) – str
- `two_value` (*optional*) – str
- `few_value` (*optional*) – str
- `many_value` (*optional*) – str

**See also:**

This object can be returned by `langpack.GetStrings`.

### 4.37.298 MaskCoords

**class** `pyrogram.api.types.MaskCoords`

**ID**

0xaed6dbb2

**Parameters**

- `n` – int 32-bit
- `x` – float 64-bit
- `y` – float 64-bit
- `zoom` – float 64-bit

### 4.37.299 Message

**class** `pyrogram.api.types.Message`

**ID**

0x44f9b43d

**Parameters**

- **id** – int 32-bit
- **to\_id** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **date** – int 32-bit
- **message** – str
- **out** (*optional*) – bool
- **mentioned** (*optional*) – bool
- **media\_unread** (*optional*) – bool
- **silent** (*optional*) – bool
- **post** (*optional*) – bool
- **from\_scheduled** (*optional*) – bool
- **from\_id** (*optional*) – int 32-bit
- **fwd\_from** (*optional*) – *MessageFwdHeader*
- **via\_bot\_id** (*optional*) – int 32-bit
- **reply\_to\_msg\_id** (*optional*) – int 32-bit
- **media** (*optional*) – Either *MessageMediaEmpty*, *MessageMediaPhoto*, *MessageMediaGeo*, *MessageMediaContact*, *MessageMediaUnsupported*, *MessageMediaDocument*, *MessageMediaWebPage*, *MessageMediaVenue*, *MessageMediaGame*, *MessageMediaInvoice*, *MessageMediaGeoLive* or *MessageMediaPoll*
- **reply\_markup** (*optional*) – Either *ReplyKeyboardHide*, *ReplyKeyboardForceReply*, *ReplyKeyboardMarkup* or *ReplyInlineMarkup*
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*
- **views** (*optional*) – int 32-bit
- **edit\_date** (*optional*) – int 32-bit
- **post\_author** (*optional*) – str
- **grouped\_id** (*optional*) – int 64-bit

**4.37.300 MessageActionBotAllowed**

```
class pyrogram.api.types.MessageActionBotAllowed
```

**ID**

0xabe9affe

**Parameters** `domain` – str

#### 4.37.301 MessageActionChannelCreate

```
class pyrogram.api.types.MessageActionChannelCreate
```

**ID**  
0x95d2ac92

**Parameters** `title` – str

#### 4.37.302 MessageActionChannelMigrateFrom

```
class pyrogram.api.types.MessageActionChannelMigrateFrom
```

**ID**  
0xb055eae

**Parameters**

- `title` – str
- `chat_id` – int 32-bit

#### 4.37.303 MessageActionChatAddUser

```
class pyrogram.api.types.MessageActionChatAddUser
```

**ID**  
0x488a7337

**Parameters** `users` – List of int 32-bit

#### 4.37.304 MessageActionChatCreate

```
class pyrogram.api.types.MessageActionChatCreate
```

**ID**  
0xa6638b9a

**Parameters**

- `title` – str
- `users` – List of int 32-bit

#### 4.37.305 MessageActionChatDeletePhoto

```
class pyrogram.api.types.MessageActionChatDeletePhoto
```

**ID**

0x95e3fbef

No parameters required.

#### 4.37.306 MessageActionChatDeleteUser

```
class pyrogram.api.types.MessageActionChatDeleteUser
```

**ID**

0xb2ae9b0c

**Parameters** `user_id` – int 32-bit

#### 4.37.307 MessageActionChatEditPhoto

```
class pyrogram.api.types.MessageActionChatEditPhoto
```

**ID**

0x7fcb13a8

**Parameters** `photo` – Either *PhotoEmpty* or *Photo*

#### 4.37.308 MessageActionChatEditTitle

```
class pyrogram.api.types.MessageActionChatEditTitle
```

**ID**

0xb5a1ce5a

**Parameters** `title` – str

#### 4.37.309 MessageActionChatJoinedByLink

```
class pyrogram.api.types.MessageActionChatJoinedByLink
```

**ID**

0xf89cf5e8

**Parameters** `inviter_id` – int 32-bit

#### 4.37.310 MessageActionChatMigrateTo

**class** pyrogram.api.types.**MessageActionChatMigrateTo**

**ID**

0x51bdb021

**Parameters** **channel\_id** – int 32-bit

#### 4.37.311 MessageActionContactSignUp

**class** pyrogram.api.types.**MessageActionContactSignUp**

**ID**

0xf3f25f76

No parameters required.

#### 4.37.312 MessageActionCustomAction

**class** pyrogram.api.types.**MessageActionCustomAction**

**ID**

0xfae69f56

**Parameters** **message** – str

#### 4.37.313 MessageActionEmpty

**class** pyrogram.api.types.**MessageActionEmpty**

**ID**

0xb6aef7b0

No parameters required.

#### 4.37.314 MessageActionGameScore

**class** pyrogram.api.types.**MessageActionGameScore**

**ID**

0x92a72876

**Parameters**

- **game\_id** – int 64-bit
- **score** – int 32-bit

#### 4.37.315 MessageActionHistoryClear

```
class pyrogram.api.types.MessageActionHistoryClear
```

ID

0x9fbab604

No parameters required.

#### 4.37.316 MessageActionPaymentSent

```
class pyrogram.api.types.MessageActionPaymentSent
```

ID

0x40699cd0

Parameters

- **currency** – str
- **total\_amount** – int 64-bit

#### 4.37.317 MessageActionPaymentSentMe

```
class pyrogram.api.types.MessageActionPaymentSentMe
```

ID

0x8f31b327

Parameters

- **currency** – str
- **total\_amount** – int 64-bit
- **payload** – bytes
- **charge** – *PaymentCharge*
- **info** (*optional*) – *PaymentRequestedInfo*
- **shipping\_option\_id** (*optional*) – str

#### 4.37.318 MessageActionPhoneCall

```
class pyrogram.api.types.MessageActionPhoneCall
```

ID

0x80e11a7f

Parameters

- **call\_id** – int 64-bit

- **reason** (*optional*) – Either *PhoneCallDiscardReasonMissed*, *PhoneCallDiscardReasonDisconnect*, *PhoneCallDiscardReasonHangup* or *PhoneCallDiscardReasonBusy*
- **duration** (*optional*) – int 32-bit

#### 4.37.319 MessageActionPinMessage

**class** pyrogram.api.types.**MessageActionPinMessage**

**ID**

0x94bd38ed

No parameters required.

#### 4.37.320 MessageActionScreenshotTaken

**class** pyrogram.api.types.**MessageActionScreenshotTaken**

**ID**

0x4792929b

No parameters required.

#### 4.37.321 MessageActionSecureValuesSent

**class** pyrogram.api.types.**MessageActionSecureValuesSent**

**ID**

0xd95c6154

**Parameters types** – List of either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*

#### 4.37.322 MessageActionSecureValuesSentMe

**class** pyrogram.api.types.**MessageActionSecureValuesSentMe**

**ID**

0x1b287353

**Parameters**

- **values** – List of *SecureValue*
- **credentials** – *SecureCredentialsEncrypted*

#### 4.37.323 MessageEmpty

```
class pyrogram.api.types.MessageEmpty
```

ID

0x83e5de54

Parameters **id** – int 32-bit

#### 4.37.324 MessageEntityBold

```
class pyrogram.api.types.MessageEntityBold
```

ID

0xbd610bc9

Parameters

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.325 MessageEntityBotCommand

```
class pyrogram.api.types.MessageEntityBotCommand
```

ID

0x6cef8ac7

Parameters

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.326 MessageEntityCashtag

```
class pyrogram.api.types.MessageEntityCashtag
```

ID

0x4c4e743f

Parameters

- **offset** – int 32-bit
- **length** – int 32-bit



#### 4.37.327 MessageEntityCode

```
class pyrogram.api.types.MessageEntityCode
```

**ID**  
0x28a20571

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.328 MessageEntityEmail

```
class pyrogram.api.types.MessageEntityEmail
```

**ID**  
0x64e475c2

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.329 MessageEntityHashtag

```
class pyrogram.api.types.MessageEntityHashtag
```

**ID**  
0x6f635b0d

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.330 MessageEntityItalic

```
class pyrogram.api.types.MessageEntityItalic
```

**ID**  
0x826f8b60

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.331 MessageEntityMention

```
class pyrogram.api.types.MessageEntityMention
```

ID

0xfa04579d

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.332 MessageEntityMentionName

```
class pyrogram.api.types.MessageEntityMentionName
```

ID

0x352dca58

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit
- **user\_id** – int 32-bit

#### 4.37.333 MessageEntityPhone

```
class pyrogram.api.types.MessageEntityPhone
```

ID

0x9b69e34b

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.334 MessageEntityPre

```
class pyrogram.api.types.MessageEntityPre
```

ID

0x73924be0

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

- **language** – str

#### 4.37.335 MessageEntityTextUrl

**class** pyrogram.api.types.**MessageEntityTextUrl**

**ID**

0x76a6d327

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit
- **url** – str

#### 4.37.336 MessageEntityUnknown

**class** pyrogram.api.types.**MessageEntityUnknown**

**ID**

0xbb92ba95

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.337 MessageEntityUrl

**class** pyrogram.api.types.**MessageEntityUrl**

**ID**

0x6ed02538

**Parameters**

- **offset** – int 32-bit
- **length** – int 32-bit

#### 4.37.338 MessageFwdHeader

**class** pyrogram.api.types.**MessageFwdHeader**

**ID**

0xec338270

**Parameters**

- **date** – int 32-bit
- **from\_id**(*optional*) – int 32-bit
- **from\_name**(*optional*) – str
- **channel\_id**(*optional*) – int 32-bit
- **channel\_post**(*optional*) – int 32-bit
- **post\_author**(*optional*) – str
- **saved\_from\_peer**(*optional*) – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **saved\_from\_msg\_id**(*optional*) – int 32-bit

#### 4.37.339 MessageMediaContact

**class** pyrogram.api.types.**MessageMediaContact**

**ID**

0xcbf24940

**Parameters**

- **phone\_number** – str
- **first\_name** – str
- **last\_name** – str
- **vcard** – str
- **user\_id** – int 32-bit

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

#### 4.37.340 MessageMediaDocument

**class** pyrogram.api.types.**MessageMediaDocument**

**ID**

0x9cb070d7

**Parameters**

- **document**(*optional*) – Either *DocumentEmpty* or *Document*
- **ttl\_seconds**(*optional*) – int 32-bit

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

#### 4.37.341 MessageMediaEmpty

**class** pyrogram.api.types.**MessageMediaEmpty**

**ID**

0x3ded6320

No parameters required.

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

#### 4.37.342 MessageMediaGame

**class** pyrogram.api.types.**MessageMediaGame**

**ID**

0xfdb19008

**Parameters** *game* – *Game*

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

#### 4.37.343 MessageMediaGeo

**class** pyrogram.api.types.**MessageMediaGeo**

**ID**

0x56e0d474

**Parameters** *geo* – Either *GeoPointEmpty* or *GeoPoint*

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

#### 4.37.344 MessageMediaGeoLive

**class** pyrogram.api.types.**MessageMediaGeoLive**

**ID**

0x7c3c2609

**Parameters**

- **geo** – Either *GeoPointEmpty* or *GeoPoint*
- **period** – int 32-bit

**See also:**

This object can be returned by `messages.GetWebPagePreview` and `messages.UploadMedia`.

### 4.37.345 MessageMediaInvoice

```
class pyrogram.api.types.MessageMediaInvoice
```

**ID**

0x84551347

**Parameters**

- **title** – str
- **description** – str
- **currency** – str
- **total\_amount** – int 64-bit
- **start\_param** – str
- **shipping\_address\_requested** (*optional*) – bool
- **test** (*optional*) – bool
- **photo** (*optional*) – Either *WebDocument* or *WebDocumentNoProxy*
- **receipt\_msg\_id** (*optional*) – int 32-bit

**See also:**

This object can be returned by `messages.GetWebPagePreview` and `messages.UploadMedia`.

### 4.37.346 MessageMediaPhoto

```
class pyrogram.api.types.MessageMediaPhoto
```

**ID**

0x695150d7

**Parameters**

- **photo** (*optional*) – Either *PhotoEmpty* or *Photo*
- **ttl\_seconds** (*optional*) – int 32-bit

**See also:**

This object can be returned by `messages.GetWebPagePreview` and `messages.UploadMedia`.

### 4.37.347 MessageMediaPoll

**class** pyrogram.api.types.**MessageMediaPoll**

**ID**

0x4bd6e798

**Parameters**

- **poll** – *Poll*
- **results** – *PollResults*

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

### 4.37.348 MessageMediaUnsupported

**class** pyrogram.api.types.**MessageMediaUnsupported**

**ID**

0x9f84f49e

No parameters required.

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

### 4.37.349 MessageMediaVenue

**class** pyrogram.api.types.**MessageMediaVenue**

**ID**

0x2ec0533f

**Parameters**

- **geo** – Either *GeoPointEmpty* or *GeoPoint*
- **title** – str
- **address** – str
- **provider** – str
- **venue\_id** – str
- **venue\_type** – str

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

### 4.37.350 MessageMediaWebPage

**class** pyrogram.api.types.**MessageMediaWebPage**

**ID**

0xa32dd600

**Parameters** **webpage** – Either *WebPageEmpty*, *WebPagePending*, *WebPage* or *WebPageNotModified*

**See also:**

This object can be returned by *messages.GetWebPagePreview* and *messages.UploadMedia*.

### 4.37.351 MessageRange

**class** pyrogram.api.types.**MessageRange**

**ID**

0x0ae30253

**Parameters**

- **min\_id** – int 32-bit
- **max\_id** – int 32-bit

**See also:**

This object can be returned by *messages.GetSplitRanges*.

### 4.37.352 MessageService

**class** pyrogram.api.types.**MessageService**

**ID**

0x9e19a1f6

**Parameters**

- **id** – int 32-bit
- **to\_id** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **date** – int 32-bit
- **action** – Either *MessageActionEmpty*, *MessageActionChatCreate*, *MessageActionChatEditTitle*, *MessageActionChatEditPhoto*, *MessageActionChatDeletePhoto*, *MessageActionChatAddUser*, *MessageActionChatDeleteUser*, *MessageActionChatJoinedByLink*, *MessageActionChannelCreate*, *MessageActionChatMigrateTo*, *MessageActionChannelMigrateFrom*, *MessageActionPinMessage*, *MessageActionHistoryClear*, *MessageActionGameScore*, *MessageActionPaymentSentMe*, *MessageActionPaymentSent*, *MessageActionPhoneCall*, *MessageActionScreenshotTaken*,



*MessageActionCustomAction, MessageActionBotAllowed,  
MessageActionSecureValuesSentMe, MessageActionSecureValuesSent  
or MessageActionContactSignUp*

- **out** (*optional*) – bool
- **mentioned** (*optional*) – bool
- **media\_unread** (*optional*) – bool
- **silent** (*optional*) – bool
- **post** (*optional*) – bool
- **from\_id** (*optional*) – int 32-bit
- **reply\_to\_msg\_id** (*optional*) – int 32-bit

#### 4.37.353 MsgDetailedInfo

**class** pyrogram.api.types.**MsgDetailedInfo**

**ID**

0x276d3ec6

**Parameters**

- **msg\_id** – int 64-bit
- **answer\_msg\_id** – int 64-bit
- **bytes** – int 32-bit
- **status** – int 32-bit

#### 4.37.354 MsgNewDetailedInfo

**class** pyrogram.api.types.**MsgNewDetailedInfo**

**ID**

0x809db6df

**Parameters**

- **answer\_msg\_id** – int 64-bit
- **bytes** – int 32-bit
- **status** – int 32-bit

#### 4.37.355 MsgResendAnsReq

**class** pyrogram.api.types.**MsgResendAnsReq**

**ID**

0x8610baeb

**Parameters** `msg_ids` – List of int 64-bit

#### 4.37.356 MsgResendReq

**class** `pyrogram.api.types.MsgResendReq`

**ID**  
0x7d861a08

**Parameters** `msg_ids` – List of int 64-bit

#### 4.37.357 MsgsAck

**class** `pyrogram.api.types.MsgsAck`

**ID**  
0x62d6b459

**Parameters** `msg_ids` – List of int 64-bit

#### 4.37.358 MsgsAllInfo

**class** `pyrogram.api.types.MsgsAllInfo`

**ID**  
0x8cc0d131

**Parameters**

- `msg_ids` – List of int 64-bit
- `info` – str

#### 4.37.359 MsgsStateInfo

**class** `pyrogram.api.types.MsgsStateInfo`

**ID**  
0x04deb57d

**Parameters**

- `req_msg_id` – int 64-bit
- `info` – str

### 4.37.360 MsgsStateReq

**class** pyrogram.api.types.MsgsStateReq

**ID**

0xda69fb52

**Parameters** `msg_ids` – List of int 64-bit

### 4.37.361 NearestDc

**class** pyrogram.api.types.NearestDc

**ID**

0x8e1a1775

**Parameters**

- **country** – str
- **this\_dc** – int 32-bit
- **nearest\_dc** – int 32-bit

**See also:**

This object can be returned by [\*help.GetNearestDc\*](#).

### 4.37.362 NewSessionCreated

**class** pyrogram.api.types.NewSessionCreated

**ID**

0x9ec20908

**Parameters**

- **first\_msg\_id** – int 64-bit
- **unique\_id** – int 64-bit
- **server\_salt** – int 64-bit

### 4.37.363 NotifyBroadcasts

**class** pyrogram.api.types.NotifyBroadcasts

**ID**

0xd612e8ef

No parameters required.

### 4.37.364 NotifyChats

**class** pyrogram.api.types.**NotifyChats**

**ID**

0xc007cec3

No parameters required.

### 4.37.365 NotifyPeer

**class** pyrogram.api.types.**NotifyPeer**

**ID**

0x9fd40bd8

**Parameters** **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*

### 4.37.366 NotifyUsers

**class** pyrogram.api.types.**NotifyUsers**

**ID**

0xb4c83b4c

No parameters required.

### 4.37.367 PQInnerData

**class** pyrogram.api.types.**PQInnerData**

**ID**

0x83c95aec

**Parameters**

- **pq** – bytes
- **p** – bytes
- **q** – bytes
- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **new\_nonce** – int 256-bit

### 4.37.368 PQInnerDataDc

**class** pyrogram.api.types.PQInnerDataDc

**ID**

0xa9f55f95

#### Parameters

- **pq** – bytes
- **p** – bytes
- **q** – bytes
- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **new\_nonce** – int 256-bit
- **dc** – int 32-bit

### 4.37.369 PQInnerDataTemp

**class** pyrogram.api.types.PQInnerDataTemp

**ID**

0x3c6a84d4

#### Parameters

- **pq** – bytes
- **p** – bytes
- **q** – bytes
- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **new\_nonce** – int 256-bit
- **expires\_in** – int 32-bit

### 4.37.370 PQInnerDataTempDc

**class** pyrogram.api.types.PQInnerDataTempDc

**ID**

0x56fddf88

#### Parameters

- **pq** – bytes
- **p** – bytes

- **q** – bytes
- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **new\_nonce** – int 256-bit
- **dc** – int 32-bit
- **expires\_in** – int 32-bit

### 4.37.371 Page

**class** pyrogram.api.types.**Page**

**ID**

0xae891bec

**Parameters**

- **url** – str
- **blocks** – List of either *PageBlockUnsupported*, *PageBlockTitle*, *PageBlockSubtitle*, *PageBlockAuthorDate*, *PageBlockHeader*, *PageBlockSubheader*, *PageBlockParagraph*, *PageBlockPreformatted*, *PageBlockFooter*, *PageBlockDivider*, *PageBlockAnchor*, *PageBlockList*, *PageBlockBlockquote*, *PageBlockPullquote*, *PageBlockPhoto*, *PageBlockVideo*, *PageBlockCover*, *PageBlockEmbed*, *PageBlockEmbedPost*, *PageBlockCollage*, *PageBlockSlideshow*, *PageBlockChannel*, *PageBlockAudio*, *PageBlockKicker*, *PageBlockTable*, *PageBlockOrderedList*, *PageBlockDetails*, *PageBlockRelatedArticles* or *PageBlockMap*
- **photos** – List of either *PhotoEmpty* or *Photo*
- **documents** – List of either *DocumentEmpty* or *Document*
- **part** (*optional*) – bool
- **rtl** (*optional*) – bool
- **v2** (*optional*) – bool

### 4.37.372 PageBlockAnchor

**class** pyrogram.api.types.**PageBlockAnchor**

**ID**

0xce0d37b0

**Parameters** **name** – str

### 4.37.373 PageBlockAudio

**class** pyrogram.api.types.**PageBlockAudio**

**ID**

0x804361ea

**Parameters**

- **audio\_id** – int 64-bit
- **caption** – *PageCaption*

### 4.37.374 PageBlockAuthorDate

**class** pyrogram.api.types.**PageBlockAuthorDate**

**ID**

0xbaafe5e0

**Parameters**

- **author** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **published\_date** – int 32-bit

### 4.37.375 PageBlockBlockquote

**class** pyrogram.api.types.**PageBlockBlockquote**

**ID**

0x263d7c26

**Parameters**

- **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **caption** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.376 PageBlockChannel

**class** pyrogram.api.types.**PageBlockChannel**

**ID**

0xef1751b5

**Parameters** **channel** – Either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*

### 4.37.377 PageBlockCollage

**class** pyrogram.api.types.**PageBlockCollage**

**ID**

0x65a0fa4d

**Parameters**

- **items** – List of either *PageBlockUnsupported*, *PageBlockTitle*, *PageBlockSubtitle*, *PageBlockAuthorDate*, *PageBlockHeader*, *PageBlockSubheader*, *PageBlockParagraph*, *PageBlockPreformatted*, *PageBlockFooter*, *PageBlockDivider*, *PageBlockAnchor*, *PageBlockList*, *PageBlockBlockquote*, *PageBlockPullquote*, *PageBlockPhoto*, *PageBlockVideo*, *PageBlockCover*, *PageBlockEmbed*, *PageBlockEmbedPost*, *PageBlockCollage*, *PageBlockSlideshow*, *PageBlockChannel*, *PageBlockAudio*, *PageBlockKicker*, *PageBlockTable*, *PageBlockOrderedList*, *PageBlockDetails*, *PageBlockRelatedArticles* or *PageBlockMap*
- **caption** – *PageCaption*

### 4.37.378 PageBlockCover

**class** pyrogram.api.types.**PageBlockCover**

**ID**

0x39f23300

**Parameters** **cover** – Either *PageBlockUnsupported*, *PageBlockTitle*, *PageBlockSubtitle*, *PageBlockAuthorDate*, *PageBlockHeader*, *PageBlockSubheader*, *PageBlockParagraph*, *PageBlockPreformatted*, *PageBlockFooter*, *PageBlockDivider*, *PageBlockAnchor*, *PageBlockList*, *PageBlockBlockquote*, *PageBlockPullquote*, *PageBlockPhoto*, *PageBlockVideo*, *PageBlockCover*, *PageBlockEmbed*, *PageBlockEmbedPost*, *PageBlockCollage*, *PageBlockSlideshow*, *PageBlockChannel*, *PageBlockAudio*, *PageBlockKicker*, *PageBlockTable*, *PageBlockOrderedList*, *PageBlockDetails*, *PageBlockRelatedArticles* or *PageBlockMap*

### 4.37.379 PageBlockDetails

**class** pyrogram.api.types.**PageBlockDetails**



**ID**

0x76768bed

**Parameters**

- **blocks** – List of either *PageBlockUnsupported*, *PageBlockTitle*, *PageBlockSubtitle*, *PageBlockAuthorDate*, *PageBlockHeader*, *PageBlockSubheader*, *PageBlockParagraph*, *PageBlockPreformatted*, *PageBlockFooter*, *PageBlockDivider*, *PageBlockAnchor*, *PageBlockList*, *PageBlockBlockquote*, *PageBlockPullquote*, *PageBlockPhoto*, *PageBlockVideo*, *PageBlockCover*, *PageBlockEmbed*, *PageBlockEmbedPost*, *PageBlockCollage*, *PageBlockSlideshow*, *PageBlockChannel*, *PageBlockAudio*, *PageBlockKicker*, *PageBlockTable*, *PageBlockOrderedList*, *PageBlockDetails*, *PageBlockRelatedArticles* or *PageBlockMap*
- **title** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **open** (*optional*) – bool

**4.37.380 PageBlockDivider****class** pyrogram.api.types.**PageBlockDivider****ID**

0xdb20b188

No parameters required.

**4.37.381 PageBlockEmbed****class** pyrogram.api.types.**PageBlockEmbed****ID**

0xa8718dc5

**Parameters**

- **caption** – *PageCaption*
- **full\_width** (*optional*) – bool
- **allow\_scrolling** (*optional*) – bool
- **url** (*optional*) – str
- **html** (*optional*) – str
- **poster\_photo\_id** (*optional*) – int 64-bit
- **w** (*optional*) – int 32-bit
- **h** (*optional*) – int 32-bit

### 4.37.382 PageBlockEmbedPost

**class** pyrogram.api.types.**PageBlockEmbedPost**

**ID**

0xf259a80b

**Parameters**

- **url** – str
- **webpage\_id** – int 64-bit
- **author\_photo\_id** – int 64-bit
- **author** – str
- **date** – int 32-bit
- **blocks** – List of either *PageBlockUnsupported*, *PageBlockTitle*, *PageBlockSubtitle*, *PageBlockAuthorDate*, *PageBlockHeader*, *PageBlockSubheader*, *PageBlockParagraph*, *PageBlockPreformatted*, *PageBlockFooter*, *PageBlockDivider*, *PageBlockAnchor*, *PageBlockList*, *PageBlockBlockquote*, *PageBlockPullquote*, *PageBlockPhoto*, *PageBlockVideo*, *PageBlockCover*, *PageBlockEmbed*, *PageBlockEmbedPost*, *PageBlockCollage*, *PageBlockSlideshow*, *PageBlockChannel*, *PageBlockAudio*, *PageBlockKicker*, *PageBlockTable*, *PageBlockOrderedList*, *PageBlockDetails*, *PageBlockRelatedArticles* or *PageBlockMap*
- **caption** – *PageCaption*

### 4.37.383 PageBlockFooter

**class** pyrogram.api.types.**PageBlockFooter**

**ID**

0x48870999

**Parameters text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.384 PageBlockHeader

**class** pyrogram.api.types.**PageBlockHeader**

**ID**

0xbfd064ec

**Parameters text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*,

*TextSubscript, TextSuperscript, TextMarked, TextPhone, TextImage or TextAnchor*

#### 4.37.385 PageBlockKicker

**class** pyrogram.api.types.**PageBlockKicker**

**ID**

0x1e148390

**Parameters** **text** – Either *TextEmpty, TextPlain, TextBold, TextItalic, TextUnderline, TextStrike, TextFixed, TextUrl, TextEmail, TextConcat, TextSubscript, TextSuperscript, TextMarked, TextPhone, TextImage or TextAnchor*

#### 4.37.386 PageBlockList

**class** pyrogram.api.types.**PageBlockList**

**ID**

0xe4e88011

**Parameters** **items** – List of either *PageListItemText* or *PageListItemBlocks*

#### 4.37.387 PageBlockMap

**class** pyrogram.api.types.**PageBlockMap**

**ID**

0xa44f3ef6

**Parameters**

- **geo** – Either *GeoPointEmpty* or *GeoPoint*
- **zoom** – int 32-bit
- **w** – int 32-bit
- **h** – int 32-bit
- **caption** – *PageCaption*

#### 4.37.388 PageBlockOrderedList

**class** pyrogram.api.types.**PageBlockOrderedList**

**ID**

0x9a8ae1e1

**Parameters** **items** – List of either *PageListOrderedItemText* or *PageListOrderedItemBlocks*

### 4.37.389 PageBlockParagraph

**class** pyrogram.api.types.**PageBlockParagraph**

**ID**

0x467a0766

**Parameters** **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.390 PageBlockPhoto

**class** pyrogram.api.types.**PageBlockPhoto**

**ID**

0x1759c560

**Parameters**

- **photo\_id** – int 64-bit
- **caption** – *PageCaption*
- **url** (*optional*) – str
- **webpage\_id** (*optional*) – int 64-bit

### 4.37.391 PageBlockPreformatted

**class** pyrogram.api.types.**PageBlockPreformatted**

**ID**

0xc070d93e

**Parameters**

- **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **language** – str

### 4.37.392 PageBlockPullquote

**class** pyrogram.api.types.**PageBlockPullquote**

**ID**

0x4f4456d3

#### Parameters

- **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **caption** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.393 PageBlockRelatedArticles

**class** pyrogram.api.types.**PageBlockRelatedArticles**

**ID**

0x16115a96

#### Parameters

- **title** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **articles** – List of *PageRelatedArticle*

### 4.37.394 PageBlockSlideshow

**class** pyrogram.api.types.**PageBlockSlideshow**

**ID**

0x031f9590

#### Parameters

- **items** – List of either *PageBlockUnsupported*, *PageBlockTitle*, *PageBlockSubtitle*, *PageBlockAuthorDate*, *PageBlockHeader*, *PageBlockSubheader*, *PageBlockParagraph*, *PageBlockPreformatted*, *PageBlockFooter*, *PageBlockDivider*, *PageBlockAnchor*, *PageBlockList*, *PageBlockBlockquote*, *PageBlockPullquote*, *PageBlockPhoto*, *PageBlockVideo*, *PageBlockCover*, *PageBlockEmbed*, *PageBlockEmbedPost*, *PageBlockCollage*, *PageBlockSlideshow*, *PageBlockChannel*, *PageBlockAudio*, *PageBlockKicker*,

*PageBlockTable, PageBlockOrderedList, PageBlockDetails,  
PageBlockRelatedArticles or PageBlockMap*

- **caption** – *PageCaption*

#### 4.37.395 PageBlockSubheader

**class** pyrogram.api.types.**PageBlockSubheader**

**ID**

0xf12bb6e1

**Parameters text** – Either *TextEmpty, TextPlain, TextBold, TextItalic, TextUnderline, TextStrike, TextFixed, TextUrl, TextEmail, TextConcat, TextSubscript, TextSuperscript, TextMarked, TextPhone, TextImage* or *TextAnchor*

#### 4.37.396 PageBlockSubtitle

**class** pyrogram.api.types.**PageBlockSubtitle**

**ID**

0x8ffa9a1f

**Parameters text** – Either *TextEmpty, TextPlain, TextBold, TextItalic, TextUnderline, TextStrike, TextFixed, TextUrl, TextEmail, TextConcat, TextSubscript, TextSuperscript, TextMarked, TextPhone, TextImage* or *TextAnchor*

#### 4.37.397 PageBlockTable

**class** pyrogram.api.types.**PageBlockTable**

**ID**

0xbf4dea82

**Parameters**

- **title** – Either *TextEmpty, TextPlain, TextBold, TextItalic, TextUnderline, TextStrike, TextFixed, TextUrl, TextEmail, TextConcat, TextSubscript, TextSuperscript, TextMarked, TextPhone, TextImage* or *TextAnchor*
- **rows** – List of *PageTableRow*
- **bordered**(*optional*) – bool
- **striped**(*optional*) – bool

### 4.37.398 PageBlockTitle

**class** pyrogram.api.types.**PageBlockTitle**

**ID**

0x70abc3fd

**Parameters** **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.399 PageBlockUnsupported

**class** pyrogram.api.types.**PageBlockUnsupported**

**ID**

0x13567e8a

No parameters required.

### 4.37.400 PageBlockVideo

**class** pyrogram.api.types.**PageBlockVideo**

**ID**

0x7c8fe7b6

**Parameters**

- **video\_id** – int 64-bit
- **caption** – *PageCaption*
- **autoplay** (*optional*) – bool
- **loop** (*optional*) – bool

### 4.37.401 PageCaption

**class** pyrogram.api.types.**PageCaption**

**ID**

0x6f747657

**Parameters**

- **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

- **credit** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

#### 4.37.402 PageListItemBlocks

**class** pyrogram.api.types.**PageListItemBlocks**

**ID**

0x25e073fc

**Parameters blocks** – List of either *PageBlockUnsupported*, *PageBlockTitle*, *PageBlockSubtitle*, *PageBlockAuthorDate*, *PageBlockHeader*, *PageBlockSubheader*, *PageBlockParagraph*, *PageBlockPreformatted*, *PageBlockFooter*, *PageBlockDivider*, *PageBlockAnchor*, *PageBlockList*, *PageBlockBlockquote*, *PageBlockPullquote*, *PageBlockPhoto*, *PageBlockVideo*, *PageBlockCover*, *PageBlockEmbed*, *PageBlockEmbedPost*, *PageBlockCollage*, *PageBlockSlideshow*, *PageBlockChannel*, *PageBlockAudio*, *PageBlockKicker*, *PageBlockTable*, *PageBlockOrderedList*, *PageBlockDetails*, *PageBlockRelatedArticles* or *PageBlockMap*

#### 4.37.403 PageListItemText

**class** pyrogram.api.types.**PageListItemText**

**ID**

0xb92fb6cd

**Parameters text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

#### 4.37.404 PageListOrderedItemBlocks

**class** pyrogram.api.types.**PageListOrderedItemBlocks**

**ID**

0x98dd8936

**Parameters**

- **num** – str
- **blocks** – List of either *PageBlockUnsupported*, *PageBlockTitle*, *PageBlockSubtitle*, *PageBlockAuthorDate*, *PageBlockHeader*, *PageBlockSubheader*, *PageBlockParagraph*, *PageBlockPreformatted*, *PageBlockFooter*, *PageBlockDivider*, *PageBlockAnchor*, *PageBlockList*, *PageBlockBlockquote*, *PageBlockPullquote*,



*PageBlockPhoto, PageBlockVideo, PageBlockCover, PageBlockEmbed, PageBlockEmbedPost, PageBlockCollage, PageBlockSlideshow, PageBlockChannel, PageBlockAudio, PageBlockKicker, PageBlockTable, PageBlockOrderedList, PageBlockDetails, PageBlockRelatedArticles or PageBlockMap*

#### 4.37.405 PageListOrderedItemText

**class** pyrogram.api.types.**PageListOrderedItemText**

**ID**

0x5e068047

**Parameters**

- **num** – str
- **text** – Either *TextEmpty, TextPlain, TextBold, TextItalic, TextUnderline, TextStrike, TextFixed, TextUrl, TextEmail, TextConcat, TextSubscript, TextSuperscript, TextMarked, TextPhone, TextImage or TextAnchor*

#### 4.37.406 PageRelatedArticle

**class** pyrogram.api.types.**PageRelatedArticle**

**ID**

0xb390dc08

**Parameters**

- **url** – str
- **webpage\_id** – int 64-bit
- **title** (*optional*) – str
- **description** (*optional*) – str
- **photo\_id** (*optional*) – int 64-bit
- **author** (*optional*) – str
- **published\_date** (*optional*) – int 32-bit

#### 4.37.407 PageTableCell

**class** pyrogram.api.types.**PageTableCell**

**ID**

0x34566b6a

**Parameters**

- **header** (*optional*) – bool

- **align\_center**(*optional*) – bool
- **align\_right**(*optional*) – bool
- **valign\_middle**(*optional*) – bool
- **valign\_bottom**(*optional*) – bool
- **text**(*optional*) – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **colspan**(*optional*) – int 32-bit
- **rowspan**(*optional*) – int 32-bit

#### 4.37.408 PageTableRow

**class** pyrogram.api.types.PageTableRow

**ID**

0xe0c0c5e5

**Parameters** **cells** – List of *PageTableCell*

#### 4.37.409 PasswordKdfAlgoSHA256SHA256PBKDF2HMACSHA512iter100000SHA256ModPow

**class** pyrogram.api.types.PasswordKdfAlgoSHA256SHA256PBKDF2HMACSHA512iter100000SHA256ModPow

**ID**

0x3a912d4a

**Parameters**

- **salt1** – bytes
- **salt2** – bytes
- **g** – int 32-bit
- **p** – bytes

#### 4.37.410 PasswordKdfAlgoUnknown

**class** pyrogram.api.types.PasswordKdfAlgoUnknown

**ID**

0xd45ab096

No parameters required.

#### 4.37.411 PaymentCharge

**class** pyrogram.api.types.PaymentCharge

**ID**

0xea02c27e

**Parameters**

- **id** – str
- **provider\_charge\_id** – str

#### 4.37.412 PaymentRequestedInfo

**class** pyrogram.api.types.PaymentRequestedInfo

**ID**

0x909c3f94

**Parameters**

- **name** (*optional*) – str
- **phone** (*optional*) – str
- **email** (*optional*) – str
- **shipping\_address** (*optional*) – *PostAddress*

#### 4.37.413 PaymentSavedCredentialsCard

**class** pyrogram.api.types.PaymentSavedCredentialsCard

**ID**

0xcdc27a1f

**Parameters**

- **id** – str
- **title** – str

#### 4.37.414 PeerChannel

**class** pyrogram.api.types.PeerChannel

**ID**

0xbddde532

**Parameters** **channel\_id** – int 32-bit

#### 4.37.415 PeerChat

**class** pyrogram.api.types.PeerChat

**ID**

0xbad0e5bb

**Parameters** `chat_id` – int 32-bit

#### 4.37.416 PeerNotifySettings

**class** pyrogram.api.types.PeerNotifySettings

**ID**

0xaf509d20

**Parameters**

- **show\_previews** (*optional*) – bool
- **silent** (*optional*) – bool
- **mute\_until** (*optional*) – int 32-bit
- **sound** (*optional*) – str

**See also:**

This object can be returned by [`account.GetNotifySettings`](#).

#### 4.37.417 PeerSettings

**class** pyrogram.api.types.PeerSettings

**ID**

0x818426cd

**Parameters** `report_spam` (*optional*) – bool

**See also:**

This object can be returned by [`messages.GetPeerSettings`](#).

#### 4.37.418 PeerUser

**class** pyrogram.api.types.PeerUser

**ID**

0x9db1bc6d

**Parameters** `user_id` – int 32-bit

#### 4.37.419 PhoneCall

```
class pyrogram.api.types.PhoneCall
```

ID

0xe6f9ddf3

##### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **date** – int 32-bit
- **admin\_id** – int 32-bit
- **participant\_id** – int 32-bit
- **g\_a\_or\_b** – bytes
- **key\_fingerprint** – int 64-bit
- **protocol** – *PhoneCallProtocol*
- **connection** – *PhoneConnection*
- **alternative\_connections** – List of *PhoneConnection*
- **start\_date** – int 32-bit
- **p2p\_allowed** (*optional*) – bool

#### 4.37.420 PhoneCallAccepted

```
class pyrogram.api.types.PhoneCallAccepted
```

ID

0x6d003d3f

##### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **date** – int 32-bit
- **admin\_id** – int 32-bit
- **participant\_id** – int 32-bit
- **g\_b** – bytes
- **protocol** – *PhoneCallProtocol*

#### 4.37.421 PhoneCallDiscardReasonBusy

```
class pyrogram.api.types.PhoneCallDiscardReasonBusy
```

**ID**

0xfaf7e8c9

No parameters required.

#### 4.37.422 PhoneCallDiscardReasonDisconnect

```
class pyrogram.api.types.PhoneCallDiscardReasonDisconnect
```

**ID**

0xe095c1a0

No parameters required.

#### 4.37.423 PhoneCallDiscardReasonHangup

```
class pyrogram.api.types.PhoneCallDiscardReasonHangup
```

**ID**

0x57adc690

No parameters required.

#### 4.37.424 PhoneCallDiscardReasonMissed

```
class pyrogram.api.types.PhoneCallDiscardReasonMissed
```

**ID**

0x85e42301

No parameters required.

#### 4.37.425 PhoneCallDiscarded

```
class pyrogram.api.types.PhoneCallDiscarded
```

**ID**

0x50ca4de1

**Parameters**

- **id** – int 64-bit
- **need\_rating** (*optional*) – bool
- **need\_debug** (*optional*) – bool

- **reason** (*optional*) – Either *PhoneCallDiscardReasonMissed*, *PhoneCallDiscardReasonDisconnect*, *PhoneCallDiscardReasonHangup* or *PhoneCallDiscardReasonBusy*
- **duration** (*optional*) – int 32-bit

#### 4.37.426 PhoneCallEmpty

**class** pyrogram.api.types.PhoneCallEmpty

**ID**

0x5366c915

**Parameters** **id** – int 64-bit

#### 4.37.427 PhoneCallProtocol

**class** pyrogram.api.types.PhoneCallProtocol

**ID**

0xa2bb35cb

**Parameters**

- **min\_layer** – int 32-bit
- **max\_layer** – int 32-bit
- **udp\_p2p** (*optional*) – bool
- **udp\_reflector** (*optional*) – bool

#### 4.37.428 PhoneCallRequested

**class** pyrogram.api.types.PhoneCallRequested

**ID**

0x83761ce4

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **date** – int 32-bit
- **admin\_id** – int 32-bit
- **participant\_id** – int 32-bit
- **g\_a\_hash** – bytes
- **protocol** – *PhoneCallProtocol*

### 4.37.429 PhoneCallWaiting

```
class pyrogram.api.types.PhoneCallWaiting
```

ID

0x1b8f4ad1

#### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **date** – int 32-bit
- **admin\_id** – int 32-bit
- **participant\_id** – int 32-bit
- **protocol** – *PhoneCallProtocol*
- **receive\_date** (*optional*) – int 32-bit

### 4.37.430 PhoneConnection

```
class pyrogram.api.types.PhoneConnection
```

ID

0x9d4c17c0

#### Parameters

- **id** – int 64-bit
- **ip** – str
- **ipv6** – str
- **port** – int 32-bit
- **peer\_tag** – bytes

### 4.37.431 Photo

```
class pyrogram.api.types.Photo
```

ID

0x9c477dd8

#### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **file\_reference** – bytes
- **date** – int 32-bit



- **sizes** – List of either *PhotoSizeEmpty*, *PhotoSize*, *PhotoCachedSize* or *PhotoStrippedSize*
- **has\_stickers** (*optional*) – bool

#### 4.37.432 PhotoCachedSize

**class** pyrogram.api.types.PhotoCachedSize

**ID**

0xe9a734fa

**Parameters**

- **type** – str
- **location** – Either *FileLocationUnavailable* or *FileLocation*
- **w** – int 32-bit
- **h** – int 32-bit
- **bytes** – bytes

#### 4.37.433 PhotoEmpty

**class** pyrogram.api.types.PhotoEmpty

**ID**

0x2331b22d

**Parameters** **id** – int 64-bit

#### 4.37.434 PhotoSize

**class** pyrogram.api.types.PhotoSize

**ID**

0x77bfb61b

**Parameters**

- **type** – str
- **location** – Either *FileLocationUnavailable* or *FileLocation*
- **w** – int 32-bit
- **h** – int 32-bit
- **size** – int 32-bit

#### 4.37.435 PhotoSizeEmpty

```
class pyrogram.api.types.PhotoSizeEmpty
```

**ID**

0x0e17e23c

**Parameters** **type** – str

#### 4.37.436 PhotoStrippedSize

```
class pyrogram.api.types.PhotoStrippedSize
```

**ID**

0xe0b0bc2e

**Parameters**

- **type** – str
- **bytes** – bytes

#### 4.37.437 Poll

```
class pyrogram.api.types.Poll
```

**ID**

0xd5529d06

**Parameters**

- **id** – int 64-bit
- **question** – str
- **answers** – List of *PollAnswer*
- **closed**(*optional*) – bool

#### 4.37.438 PollAnswer

```
class pyrogram.api.types.PollAnswer
```

**ID**

0x6ca9c2e9

**Parameters**

- **text** – str
- **option** – bytes

#### 4.37.439 PollAnswerVoters

**class** pyrogram.api.types.PollAnswerVoters

**ID**

0x3b6ddad2

**Parameters**

- **option** – bytes
- **voters** – int 32-bit
- **chosen** (*optional*) – bool

#### 4.37.440 PollResults

**class** pyrogram.api.types.PollResults

**ID**

0x5755785a

**Parameters**

- **min** (*optional*) – bool
- **results** (*optional*) – List of [PollAnswerVoters](#)
- **total\_voters** (*optional*) – int 32-bit

#### 4.37.441 Pong

**class** pyrogram.api.types.Pong

**ID**

0x347773c5

**Parameters**

- **msg\_id** – int 64-bit
- **ping\_id** – int 64-bit

**See also:**

This object can be returned by [Ping](#) and [PingDelayDisconnect](#).

#### 4.37.442 PopularContact

**class** pyrogram.api.types.PopularContact

**ID**

0x5ce14175

**Parameters**

- **client\_id** – int 64-bit
- **importers** – int 32-bit

#### 4.37.443 PostAddress

```
class pyrogram.api.types.PostAddress
```

**ID**

0x1e8caaeb

**Parameters**

- **street\_line1** – str
- **street\_line2** – str
- **city** – str
- **state** – str
- **country\_iso2** – str
- **post\_code** – str

#### 4.37.444 PrivacyKeyChatInvite

```
class pyrogram.api.types.PrivacyKeyChatInvite
```

**ID**

0x500e6dfa

No parameters required.

#### 4.37.445 PrivacyKeyForwards

```
class pyrogram.api.types.PrivacyKeyForwards
```

**ID**

0x69ec56a3

No parameters required.

#### 4.37.446 PrivacyKeyPhoneCall

```
class pyrogram.api.types.PrivacyKeyPhoneCall
```

**ID**

0x3d662b7b

No parameters required.

#### 4.37.447 PrivacyKeyPhoneP2P

```
class pyrogram.api.types.PrivacyKeyPhoneP2P
```

**ID**

0x39491cc8

No parameters required.

#### 4.37.448 PrivacyKeyProfilePhoto

```
class pyrogram.api.types.PrivacyKeyProfilePhoto
```

**ID**

0x96151fed

No parameters required.

#### 4.37.449 PrivacyKeyStatusTimestamp

```
class pyrogram.api.types.PrivacyKeyStatusTimestamp
```

**ID**

0xbc2eab30

No parameters required.

#### 4.37.450 PrivacyValueAllowAll

```
class pyrogram.api.types.PrivacyValueAllowAll
```

**ID**

0x65427b82

No parameters required.

#### 4.37.451 PrivacyValueAllowContacts

```
class pyrogram.api.types.PrivacyValueAllowContacts
```

**ID**

0xffffelbac

No parameters required.

#### 4.37.452 PrivacyValueAllowUsers

**class** pyrogram.api.types.PrivacyValueAllowUsers

**ID**

0x4d5bbe0c

**Parameters** **users** – List of int 32-bit

#### 4.37.453 PrivacyValueDisallowAll

**class** pyrogram.api.types.PrivacyValueDisallowAll

**ID**

0x8b73e763

No parameters required.

#### 4.37.454 PrivacyValueDisallowContacts

**class** pyrogram.api.types.PrivacyValueDisallowContacts

**ID**

0xf888fa1a

No parameters required.

#### 4.37.455 PrivacyValueDisallowUsers

**class** pyrogram.api.types.PrivacyValueDisallowUsers

**ID**

0x0c7f49b7

**Parameters** **users** – List of int 32-bit

#### 4.37.456 ReceivedNotifyMessage

**class** pyrogram.api.types.ReceivedNotifyMessage

**ID**

0xa384b779

**Parameters**

- **id** – int 32-bit
- **flags** – int 32-bit

**See also:**

This object can be returned by `messages.ReceivedMessages`.

**4.37.457 RecentMeUrlChat**

```
class pyrogram.api.types.RecentMeUrlChat
```

**ID**

0xa01b22f9

**Parameters**

- **url** – str
- **chat\_id** – int 32-bit

**4.37.458 RecentMeUrlChatInvite**

```
class pyrogram.api.types.RecentMeUrlChatInvite
```

**ID**

0xeb49081d

**Parameters**

- **url** – str
- **chat\_invite** – Either *ChatInviteAlready* or *ChatInvite*

**4.37.459 RecentMeUrlStickerSet**

```
class pyrogram.api.types.RecentMeUrlStickerSet
```

**ID**

0xbc0a57dc

**Parameters**

- **url** – str
- **set** – Either *StickerSetCovered* or *StickerSetMultiCovered*

**4.37.460 RecentMeUrlUnknown**

```
class pyrogram.api.types.RecentMeUrlUnknown
```

**ID**

0x46e1d13d

**Parameters** **url** – str

#### 4.37.461 RecentMeUrlUser

**class** pyrogram.api.types.RecentMeUrlUser

**ID**  
0x8dbc3336

**Parameters**

- **url** – str
- **user\_id** – int 32-bit

#### 4.37.462 ReplyInlineMarkup

**class** pyrogram.api.types.ReplyInlineMarkup

**ID**  
0x48a30254

**Parameters** **rows** – List of *KeyboardButtonRow*

#### 4.37.463 ReplyKeyboardForceReply

**class** pyrogram.api.types.ReplyKeyboardForceReply

**ID**  
0xf4108aa0

**Parameters**

- **single\_use** (*optional*) – bool
- **selective** (*optional*) – bool

#### 4.37.464 ReplyKeyboardHide

**class** pyrogram.api.types.ReplyKeyboardHide

**ID**  
0xa03e5b85

**Parameters** **selective** (*optional*) – bool

#### 4.37.465 ReplyKeyboardMarkup

**class** pyrogram.api.types.ReplyKeyboardMarkup

**ID**  
0x3502758c



**Parameters**

- **rows** – List of [KeyboardButtonRow](#)
- **resize**(*optional*) – bool
- **single\_use**(*optional*) – bool
- **selective**(*optional*) – bool

**4.37.466 ResPQ**

```
class pyrogram.api.types.ResPQ
```

**ID**

```
0x05162463
```

**Parameters**

- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **pq** – bytes
- **server\_public\_key\_fingerprints** – List of int 64-bit

**See also:**

This object can be returned by [ReqPq](#) and [ReqPqMulti](#).

**4.37.467 RpcAnswerDropped**

```
class pyrogram.api.types.RpcAnswerDropped
```

**ID**

```
0xa43ad8b7
```

**Parameters**

- **msg\_id** – int 64-bit
- **seq\_no** – int 32-bit
- **bytes** – int 32-bit

**See also:**

This object can be returned by [RpcDropAnswer](#).

**4.37.468 RpcAnswerDroppedRunning**

```
class pyrogram.api.types.RpcAnswerDroppedRunning
```

**ID**

```
0xcd78e586
```

No parameters required.

**See also:**

This object can be returned by [\*RpcDropAnswer\*](#).

#### 4.37.469 RpcAnswerUnknown

**class** `pyrogram.api.types.RpcAnswerUnknown`

**ID**

0x5e2ad36e

No parameters required.

**See also:**

This object can be returned by [\*RpcDropAnswer\*](#).

#### 4.37.470 RpcError

**class** `pyrogram.api.types.RpcError`

**ID**

0x2144ca19

**Parameters**

- **error\_code** – int 32-bit
- **error\_message** – str

#### 4.37.471 RpcResult

**class** `pyrogram.api.types.RpcResult`

**ID**

0xf35c6d01

**Parameters**

- **req\_msg\_id** – int 64-bit
- **result** – Any object from [\*pyrogram.api.types\*](#)

#### 4.37.472 SavedPhoneContact

**class** `pyrogram.api.types.SavedPhoneContact`

**ID**

0x1142bd56

**Parameters**

- **phone** – str
- **first\_name** – str
- **last\_name** – str
- **date** – int 32-bit

See also:

This object can be returned by `contacts.GetSaved`.

#### 4.37.473 SecureCredentialsEncrypted

```
class pyrogram.api.types.SecureCredentialsEncrypted
```

**ID**  
0x33f0ea47

##### Parameters

- **data** – bytes
- **hash** – bytes
- **secret** – bytes

#### 4.37.474 SecureData

```
class pyrogram.api.types.SecureData
```

**ID**  
0x8aeabec3

##### Parameters

- **data** – bytes
- **data\_hash** – bytes
- **secret** – bytes

#### 4.37.475 SecureFile

```
class pyrogram.api.types.SecureFile
```

**ID**  
0xe0277a62

##### Parameters

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **size** – int 32-bit

- **dc\_id** – int 32-bit
- **date** – int 32-bit
- **file\_hash** – bytes
- **secret** – bytes

#### 4.37.476 SecureFileEmpty

```
class pyrogram.api.types.SecureFileEmpty
```

**ID**

0x64199744

No parameters required.

#### 4.37.477 SecurePasswordKdfAlgoPBKDF2HMACSHA512iter100000

```
class pyrogram.api.types.SecurePasswordKdfAlgoPBKDF2HMACSHA512iter100000
```

**ID**

0xbbf2dda0

**Parameters** **salt** – bytes

#### 4.37.478 SecurePasswordKdfAlgoSHA512

```
class pyrogram.api.types.SecurePasswordKdfAlgoSHA512
```

**ID**

0x86471d92

**Parameters** **salt** – bytes

#### 4.37.479 SecurePasswordKdfAlgoUnknown

```
class pyrogram.api.types.SecurePasswordKdfAlgoUnknown
```

**ID**

0x004a8537

No parameters required.

#### 4.37.480 SecurePlainEmail

```
class pyrogram.api.types.SecurePlainEmail
```

**ID**

0x21ec5a5f

**Parameters** `email` – str

#### 4.37.481 SecurePlainPhone

**class** `pyrogram.api.types.SecurePlainPhone`

**ID**  
0x7d6099dd

**Parameters** `phone` – str

#### 4.37.482 SecureRequiredType

**class** `pyrogram.api.types.SecureRequiredType`

**ID**  
0x829d99da

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **native\_names** (*optional*) – bool
- **selfie\_required** (*optional*) – bool
- **translation\_required** (*optional*) – bool

#### 4.37.483 SecureRequiredTypeOneOf

**class** `pyrogram.api.types.SecureRequiredTypeOneOf`

**ID**  
0x027477b4

**Parameters** `types` – List of either *SecureRequiredType* or *SecureRequiredTypeOneOf*

#### 4.37.484 SecureSecretSettings

**class** `pyrogram.api.types.SecureSecretSettings`

**ID**  
0x1527bcac

**Parameters**

- **secure\_algo** – Either *SecurePasswordKdfAlgoUnknown*, *SecurePasswordKdfAlgoPBKDF2HMACSHA512iter100000* or *SecurePasswordKdfAlgoSHA512*
- **secure\_secret** – bytes
- **secure\_secret\_id** – int 64-bit

### 4.37.485 SecureValue

```
class pyrogram.api.types.SecureValue
```

**ID**

0x187fa0ca

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **hash** – bytes
- **data** (optional) – *SecureData*
- **front\_side** (optional) – Either *SecureFileEmpty* or *SecureFile*
- **reverse\_side** (optional) – Either *SecureFileEmpty* or *SecureFile*
- **selfie** (optional) – Either *SecureFileEmpty* or *SecureFile*
- **translation** (optional) – List of either *SecureFileEmpty* or *SecureFile*
- **files** (optional) – List of either *SecureFileEmpty* or *SecureFile*
- **plain\_data** (optional) – Either *SecurePlainPhone* or *SecurePlainEmail*

**See also:**

This object can be returned by *account.GetAllSecureValues*, *account.GetSecureValue* and *account.SaveSecureValue*.

### 4.37.486 SecureValueError

```
class pyrogram.api.types.SecureValueError
```

**ID**

0x869d758f

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **hash** – bytes
- **text** – str

#### 4.37.487 SecureValueErrorData

**class** pyrogram.api.types.**SecureValueErrorData**

**ID**

0xe8a40bd9

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **data\_hash** – bytes
- **field** – str
- **text** – str

#### 4.37.488 SecureValueErrorFile

**class** pyrogram.api.types.**SecureValueErrorFile**

**ID**

0x7a700873

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **file\_hash** – bytes
- **text** – str

#### 4.37.489 SecureValueErrorFiles

**class** pyrogram.api.types.SecureValueErrorFiles

**ID**

0x666220e9

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **file\_hash** – List of bytes
- **text** – str

#### 4.37.490 SecureValueErrorFrontSide

**class** pyrogram.api.types.SecureValueErrorFrontSide

**ID**

0x00be3dfa

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **file\_hash** – bytes
- **text** – str

#### 4.37.491 SecureValueErrorReverseSide

**class** pyrogram.api.types.SecureValueErrorReverseSide

**ID**

0x868a2aa5

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*,



*SecureValueTypeAddress, SecureValueTypeUtilityBill,  
SecureValueTypeBankStatement, SecureValueTypeRentalAgreement,  
SecureValueTypePassportRegistration, SecureValueTypeTemporaryRegistration,  
SecureValueTypePhone or SecureValueTypeEmail*

- **file\_hash** – bytes
- **text** – str

#### 4.37.492 SecureValueErrorSelfie

**class** pyrogram.api.types.**SecureValueErrorSelfie**

**ID**

0xe537ced6

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails, SecureValueTypePassport, SecureValueTypeDriverLicense, SecureValueTypeIdentityCard, SecureValueTypeInternalPassport, SecureValueTypeAddress, SecureValueTypeUtilityBill, SecureValueTypeBankStatement, SecureValueTypeRentalAgreement, SecureValueTypePassportRegistration, SecureValueTypeTemporaryRegistration, SecureValueTypePhone or SecureValueTypeEmail*
- **file\_hash** – bytes
- **text** – str

#### 4.37.493 SecureValueErrorTranslationFile

**class** pyrogram.api.types.**SecureValueErrorTranslationFile**

**ID**

0xa1144770

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails, SecureValueTypePassport, SecureValueTypeDriverLicense, SecureValueTypeIdentityCard, SecureValueTypeInternalPassport, SecureValueTypeAddress, SecureValueTypeUtilityBill, SecureValueTypeBankStatement, SecureValueTypeRentalAgreement, SecureValueTypePassportRegistration, SecureValueTypeTemporaryRegistration, SecureValueTypePhone or SecureValueTypeEmail*
- **file\_hash** – bytes
- **text** – str

#### 4.37.494 SecureValueErrorTranslationFiles

**class** pyrogram.api.types.SecureValueErrorTranslationFiles

**ID**

0x34636dd8

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **file\_hash** – List of bytes
- **text** – str

#### 4.37.495 SecureValueHash

**class** pyrogram.api.types.SecureValueHash

**ID**

0xed1ecdb0

**Parameters**

- **type** – Either *SecureValueTypePersonalDetails*, *SecureValueTypePassport*, *SecureValueTypeDriverLicense*, *SecureValueTypeIdentityCard*, *SecureValueTypeInternalPassport*, *SecureValueTypeAddress*, *SecureValueTypeUtilityBill*, *SecureValueTypeBankStatement*, *SecureValueTypeRentalAgreement*, *SecureValueTypePassportRegistration*, *SecureValueTypeTemporaryRegistration*, *SecureValueTypePhone* or *SecureValueTypeEmail*
- **hash** – bytes

#### 4.37.496 SecureValueTypeAddress

**class** pyrogram.api.types.SecureValueTypeAddress

**ID**

0xcbe31e26

No parameters required.

#### 4.37.497 SecureValueTypeBankStatement

**class** pyrogram.api.types.SecureValueTypeBankStatement

**ID**

0x89137c0d

No parameters required.

#### 4.37.498 SecureValueTypeDriverLicense

```
class pyrogram.api.types.SecureValueTypeDriverLicense
```

**ID**

0x06e425c4

No parameters required.

#### 4.37.499 SecureValueTypeEmail

```
class pyrogram.api.types.SecureValueTypeEmail
```

**ID**

0x8e3ca7ee

No parameters required.

#### 4.37.500 SecureValueTypeIdentityCard

```
class pyrogram.api.types.SecureValueTypeIdentityCard
```

**ID**

0xa0d0744b

No parameters required.

#### 4.37.501 SecureValueTypeInternalPassport

```
class pyrogram.api.types.SecureValueTypeInternalPassport
```

**ID**

0x99a48f23

No parameters required.

#### 4.37.502 SecureValueTypePassport

```
class pyrogram.api.types.SecureValueTypePassport
```

**ID**

0x3dac6a00

No parameters required.

#### 4.37.503 SecureValueTypePassportRegistration

```
class pyrogram.api.types.SecureValueTypePassportRegistration
```

**ID**

0x99e3806a

No parameters required.

#### 4.37.504 SecureValueTypePersonalDetails

```
class pyrogram.api.types.SecureValueTypePersonalDetails
```

**ID**

0x9d2a81e3

No parameters required.

#### 4.37.505 SecureValueTypePhone

```
class pyrogram.api.types.SecureValueTypePhone
```

**ID**

0xb320aadb

No parameters required.

#### 4.37.506 SecureValueTypeRentalAgreement

```
class pyrogram.api.types.SecureValueTypeRentalAgreement
```

**ID**

0x8b883488

No parameters required.

#### 4.37.507 SecureValueTypeTemporaryRegistration

```
class pyrogram.api.types.SecureValueTypeTemporaryRegistration
```

**ID**

0xea02ec33

No parameters required.

#### 4.37.508 SecureValueTypeUtilityBill

```
class pyrogram.api.types.SecureValueTypeUtilityBill
```

**ID**

0xfc36954e

No parameters required.

#### 4.37.509 SendMessageCancelAction

```
class pyrogram.api.types.SendMessageCancelAction
```

**ID**

0xfd5ec8f5

No parameters required.

#### 4.37.510 SendMessageChooseContactAction

```
class pyrogram.api.types.SendMessageChooseContactAction
```

**ID**

0x628cbc6f

No parameters required.

#### 4.37.511 SendMessageGamePlayAction

```
class pyrogram.api.types.SendMessageGamePlayAction
```

**ID**

0xdd6a8f48

No parameters required.

#### 4.37.512 SendMessageGeoLocationAction

```
class pyrogram.api.types.SendMessageGeoLocationAction
```

**ID**

0x176f8ba1

No parameters required.

#### 4.37.513 SendMessageRecordAudioAction

```
class pyrogram.api.types.SendMessageRecordAudioAction
```

**ID**

0xd52f73f7

No parameters required.

#### 4.37.514 SendMessageRecordRoundAction

```
class pyrogram.api.types.SendMessageRecordRoundAction
```

**ID**

0x88f27fbc

No parameters required.

#### 4.37.515 SendMessageRecordVideoAction

```
class pyrogram.api.types.SendMessageRecordVideoAction
```

**ID**

0xa187d66f

No parameters required.

#### 4.37.516 SendMessageTypingAction

```
class pyrogram.api.types.SendMessageTypingAction
```

**ID**

0x16bf744e

No parameters required.

#### 4.37.517 SendMessageUploadAudioAction

```
class pyrogram.api.types.SendMessageUploadAudioAction
```

**ID**

0xf351d7ab

**Parameters** `progress` – int 32-bit

#### 4.37.518 SendMessageUploadDocumentAction

```
class pyrogram.api.types.SendMessageUploadDocumentAction
```

**ID**

0xaa0cd9e4

**Parameters** `progress` – int 32-bit

#### 4.37.519 SendMessageUploadPhotoAction

```
class pyrogram.api.types.SendMessageUploadPhotoAction
```

**ID**

0xd1d34a26

**Parameters** `progress` – int 32-bit

#### 4.37.520 SendMessageUploadRoundAction

```
class pyrogram.api.types.SendMessageUploadRoundAction
```

**ID**

0x243e1c66

**Parameters** `progress` – int 32-bit

#### 4.37.521 SendMessageUploadVideoAction

```
class pyrogram.api.types.SendMessageUploadVideoAction
```

**ID**

0xe9763aec

**Parameters** `progress` – int 32-bit

#### 4.37.522 ServerDhInnerData

```
class pyrogram.api.types.ServerDhInnerData
```

**ID**

0xb5890dba

**Parameters**

- `nonce` – int 128-bit
- `server_nonce` – int 128-bit

- **g** – int 32-bit
- **dh\_prime** – bytes
- **g\_a** – bytes
- **server\_time** – int 32-bit

#### 4.37.523 ServerDHParamsFail

**class** pyrogram.api.types.**ServerDHParamsFail**

**ID**

0x79cb045d

**Parameters**

- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **new\_nonce\_hash** – int 128-bit

**See also:**

This object can be returned by [ReqDHParams](#).

#### 4.37.524 ServerDHParamsOk

**class** pyrogram.api.types.**ServerDHParamsOk**

**ID**

0xd0e8075c

**Parameters**

- **nonce** – int 128-bit
- **server\_nonce** – int 128-bit
- **encrypted\_answer** – bytes

**See also:**

This object can be returned by [ReqDHParams](#).

#### 4.37.525 ShippingOption

**class** pyrogram.api.types.**ShippingOption**

**ID**

0xb6213cdf

**Parameters**

- **id** – str



- **title** – str
- **prices** – List of *LabeledPrice*

#### 4.37.526 StatsURL

**class** pyrogram.api.types.StatsURL

**ID**

0x47a971e0

**Parameters** **url** – str

**See also:**

This object can be returned by *messages.GetStatsURL*.

#### 4.37.527 StickerPack

**class** pyrogram.api.types.StickerPack

**ID**

0x12b299d4

**Parameters**

- **emoticon** – str
- **documents** – List of int 64-bit

#### 4.37.528 StickerSet

**class** pyrogram.api.types.StickerSet

**ID**

0x6a90bcb7

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **title** – str
- **short\_name** – str
- **count** – int 32-bit
- **hash** – int 32-bit
- **archived** (*optional*) – bool
- **official** (*optional*) – bool
- **masks** (*optional*) – bool

- **installed\_date** (*optional*) – int 32-bit
- **thumb** (*optional*) – Either *PhotoSizeEmpty*, *PhotoSize*, *PhotoCachedSize* or *PhotoStrippedSize*

#### 4.37.529 StickerSetCovered

**class** pyrogram.api.types.StickerSetCovered

**ID**

0x6410a5d2

**Parameters**

- **set** – *StickerSet*
- **cover** – Either *DocumentEmpty* or *Document*

**See also:**

This object can be returned by *messages.GetAttachedStickers*.

#### 4.37.530 StickerSetMultiCovered

**class** pyrogram.api.types.StickerSetMultiCovered

**ID**

0x3407e51b

**Parameters**

- **set** – *StickerSet*
- **covers** – List of either *DocumentEmpty* or *Document*

**See also:**

This object can be returned by *messages.GetAttachedStickers*.

#### 4.37.531 TextAnchor

**class** pyrogram.api.types.TextAnchor

**ID**

0x35553762

**Parameters**

- **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **name** – str

### 4.37.532 TextBold

**class** pyrogram.api.types.**TextBold**

**ID**

0x6724abc4

**Parameters** **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.533 TextConcat

**class** pyrogram.api.types.**TextConcat**

**ID**

0x7e6260d7

**Parameters** **texts** – List of either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.534 TextEmail

**class** pyrogram.api.types.**TextEmail**

**ID**

0xde5a0dd6

**Parameters**

- **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **email** – str

### 4.37.535 TextEmpty

**class** pyrogram.api.types.**TextEmpty**

**ID**

0xdc3d824f

No parameters required.

### 4.37.536 TextFixed

**class** pyrogram.api.types.**TextFixed**

**ID**

0x6c3f19b9

**Parameters text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.537 TextImage

**class** pyrogram.api.types.**TextImage**

**ID**

0x081ccf4f

**Parameters**

- **document\_id** – int 64-bit
- **w** – int 32-bit
- **h** – int 32-bit

### 4.37.538 TextItalic

**class** pyrogram.api.types.**TextItalic**

**ID**

0xd912a59c

**Parameters text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.539 TextMarked

**class** pyrogram.api.types.**TextMarked**

**ID**

0x034b8621

**Parameters text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.540 TextPhone

**class** pyrogram.api.types.**TextPhone**

**ID**

0x1ccb966a

**Parameters**

- **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **phone** – str

### 4.37.541 TextPlain

**class** pyrogram.api.types.**TextPlain**

**ID**

0x744694e0

**Parameters** **text** – str

### 4.37.542 TextStrike

**class** pyrogram.api.types.**TextStrike**

**ID**

0x9bf8bb95

**Parameters** **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

### 4.37.543 TextSubscript

**class** pyrogram.api.types.**TextSubscript**

**ID**

0xed6a8504

**Parameters** **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

#### 4.37.544 TextSuperscript

**class** pyrogram.api.types.**TextSuperscript**

**ID**

0xc7fb5e01

**Parameters text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

#### 4.37.545 TextUnderline

**class** pyrogram.api.types.**TextUnderline**

**ID**

0xc12622c4

**Parameters text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*

#### 4.37.546 TextUrl

**class** pyrogram.api.types.**TextUrl**

**ID**

0x3c2884c1

**Parameters**

- **text** – Either *TextEmpty*, *TextPlain*, *TextBold*, *TextItalic*, *TextUnderline*, *TextStrike*, *TextFixed*, *TextUrl*, *TextEmail*, *TextConcat*, *TextSubscript*, *TextSuperscript*, *TextMarked*, *TextPhone*, *TextImage* or *TextAnchor*
- **url** – str
- **webpage\_id** – int 64-bit

#### 4.37.547 TopPeer

**class** pyrogram.api.types.**TopPeer**

**ID**

0xedcdc05b

**Parameters**

- **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **rating** – float 64-bit

#### 4.37.548 TopPeerCategoryBotsInline

**class** pyrogram.api.types.TopPeerCategoryBotsInline

**ID**

0x148677e2

No parameters required.

#### 4.37.549 TopPeerCategoryBotsPM

**class** pyrogram.api.types.TopPeerCategoryBotsPM

**ID**

0xab661b5b

No parameters required.

#### 4.37.550 TopPeerCategoryChannels

**class** pyrogram.api.types.TopPeerCategoryChannels

**ID**

0x161d9628

No parameters required.

#### 4.37.551 TopPeerCategoryCorrespondents

**class** pyrogram.api.types.TopPeerCategoryCorrespondents

**ID**

0x0637b7ed

No parameters required.

#### 4.37.552 TopPeerCategoryGroups

**class** pyrogram.api.types.TopPeerCategoryGroups

**ID**

0xbd17a14a

No parameters required.

### 4.37.553 TopPeerCategoryPeers

**class** pyrogram.api.types.TopPeerCategoryPeers

**ID**

0xfb834291

**Parameters**

- **category** – Either *TopPeerCategoryBotsPM*, *TopPeerCategoryBotsInline*, *TopPeerCategoryCorrespondents*, *TopPeerCategoryGroups*, *TopPeerCategoryChannels* or *TopPeerCategoryPhoneCalls*
- **count** – int 32-bit
- **peers** – List of *TopPeer*

### 4.37.554 TopPeerCategoryPhoneCalls

**class** pyrogram.api.types.TopPeerCategoryPhoneCalls

**ID**

0x1e76a78c

No parameters required.

### 4.37.555 Update

**class** pyrogram.api.types.Update

**ID**

0x74ae4240

**Parameters**

- **updates** – List of either *UpdateNewMessage*, *UpdateMessageID*, *UpdateDeleteMessages*, *UpdateUserTyping*, *UpdateChatUserTyping*, *UpdateChatParticipants*, *UpdateUserStatus*, *UpdateUserName*, *UpdateUserPhoto*, *UpdateContactLink*, *UpdateNewEncryptedMessage*, *UpdateEncryptedChatTyping*, *UpdateEncryption*, *UpdateEncryptedMessagesRead*, *UpdateChatParticipantAdd*, *UpdateChatParticipantDelete*, *UpdateDcOptions*, *UpdateUserBlocked*, *UpdateNotifySettings*, *UpdateServiceNotification*, *UpdatePrivacy*, *UpdateUserPhone*, *UpdateReadHistoryInbox*, *UpdateReadHistoryOutbox*, *UpdateWebPage*, *UpdateReadMessagesContents*, *UpdateChannelTooLong*, *UpdateChannel*, *UpdateNewChannelMessage*, *UpdateReadChannelInbox*, *UpdateDeleteChannelMessages*, *UpdateChannelMessageViews*, *UpdateChatParticipantAdmin*, *UpdateNewStickerSet*, *UpdateStickerSetsOrder*, *UpdateStickerSets*, *UpdateSavedGifs*, *UpdateBotInlineQuery*, *UpdateBotInlineSend*, *UpdateEditChannelMessage*, *UpdateChannelPinnedMessage*, *UpdateBotCallbackQuery*, *UpdateEditMessage*,



```

UpdateInlineBotCallbackQuery, UpdateReadChannelOutbox,
UpdateDraftMessage, UpdateReadFeaturedStickers,
UpdateRecentStickers, UpdateConfig, UpdatePtsChanged,
UpdateChannelWebPage, UpdateDialogPinned, UpdatePinnedDialogs,
UpdateBotWebhookJSON, UpdateBotWebhookJSONQuery,
UpdateBotShippingQuery, UpdateBotPrecheckoutQuery,
UpdatePhoneCall, UpdateLangPackTooLong, UpdateLangPack,
UpdateFavedStickers, UpdateChannelReadMessagesContents,
UpdateContactsReset, UpdateChannelAvailableMessages,
UpdateDialogUnreadMark, UpdateUserPinnedMessage,
UpdateChatPinnedMessage, UpdateMessagePoll or
UpdateChatDefaultBannedRights

```

- **users** – List of either *UserEmpty* or *User*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*
- **date** – int 32-bit
- **seq** – int 32-bit

#### See also:

This object can be returned by *account.GetNotifyExceptions*, *messages.SendMessage*, *messages.SendMedia*, *messages.ForwardMessages*, *messages.EditChatTitle*, *messages.EditChatPhoto*, *messages.AddChatUser*, *messages.DeleteChatUser*, *messages.CreateChat*, *messages.ImportChatInvite*, *messages.StartBot*, *messages.MigrateChat*, *messages.SendInlineBotResult*, *messages.EditMessage*, *messages.GetAllDrafts*, *messages.SetGameScore*, *messages.SendScreenshotNotification*, *messages.SendMultiMedia*, *messages.UpdatePinnedMessage*, *messages.SendVote*, *messages.GetPollResults*, *messages.EditChatDefaultBannedRights*, *help.GetAppChangelog*, *channels.CreateChannel*, *channels.EditAdmin*, *channels.EditTitle*, *channels.EditPhoto*, *channels.JoinChannel*, *channels.LeaveChannel*, *channels.InviteToChannel*, *channels.DeleteChannel*, *channels.ToggleSignatures*, *channels.EditBanned*, *channels.TogglePreHistoryHidden*, *phone.DiscardCall* and *phone.SetCallRating*.

## 4.37.556 UpdateBotCallbackQuery

**class** pyrogram.api.types.UpdateBotCallbackQuery

#### ID

0xe73547e1

#### Parameters

- **query\_id** – int 64-bit
- **user\_id** – int 32-bit
- **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **msg\_id** – int 32-bit
- **chat\_instance** – int 64-bit
- **data** (*optional*) – bytes

- **game\_short\_name** (*optional*) – str

#### 4.37.557 UpdateBotInlineQuery

**class** pyrogram.api.types.UpdateBotInlineQuery

**ID**

0x54826690

**Parameters**

- **query\_id** – int 64-bit
- **user\_id** – int 32-bit
- **query** – str
- **offset** – str
- **geo** (*optional*) – Either *GeoPointEmpty* or *GeoPoint*

#### 4.37.558 UpdateBotInlineSend

**class** pyrogram.api.types.UpdateBotInlineSend

**ID**

0x0e48f964

**Parameters**

- **user\_id** – int 32-bit
- **query** – str
- **id** – str
- **geo** (*optional*) – Either *GeoPointEmpty* or *GeoPoint*
- **msg\_id** (*optional*) – *InputBotInlineMessageID*

#### 4.37.559 UpdateBotPrecheckoutQuery

**class** pyrogram.api.types.UpdateBotPrecheckoutQuery

**ID**

0x5d2f3aa9

**Parameters**

- **query\_id** – int 64-bit
- **user\_id** – int 32-bit
- **payload** – bytes
- **currency** – str

- **total\_amount** – int 64-bit
- **info** (*optional*) – *PaymentRequestedInfo*
- **shipping\_option\_id** (*optional*) – str

#### 4.37.560 UpdateBotShippingQuery

**class** pyrogram.api.types.UpdateBotShippingQuery

**ID**

0xe0cdc940

**Parameters**

- **query\_id** – int 64-bit
- **user\_id** – int 32-bit
- **payload** – bytes
- **shipping\_address** – *PostAddress*

#### 4.37.561 UpdateBotWebhookJSON

**class** pyrogram.api.types.UpdateBotWebhookJSON

**ID**

0x8317c0c3

**Parameters** **data** – *DataJSON*

#### 4.37.562 UpdateBotWebhookJSONQuery

**class** pyrogram.api.types.UpdateBotWebhookJSONQuery

**ID**

0x9b9240a6

**Parameters**

- **query\_id** – int 64-bit
- **data** – *DataJSON*
- **timeout** – int 32-bit

#### 4.37.563 UpdateChannel

**class** pyrogram.api.types.UpdateChannel

**ID**

0xb6d45656

Parameters **channel\_id** – int 32-bit

#### 4.37.564 UpdateChannelAvailableMessages

**class** pyrogram.api.types.UpdateChannelAvailableMessages

**ID**  
0x70db6837

**Parameters**

- **channel\_id** – int 32-bit
- **available\_min\_id** – int 32-bit

#### 4.37.565 UpdateChannelMessageViews

**class** pyrogram.api.types.UpdateChannelMessageViews

**ID**  
0x98a12b4b

**Parameters**

- **channel\_id** – int 32-bit
- **id** – int 32-bit
- **views** – int 32-bit

#### 4.37.566 UpdateChannelPinnedMessage

**class** pyrogram.api.types.UpdateChannelPinnedMessage

**ID**  
0x98592475

**Parameters**

- **channel\_id** – int 32-bit
- **id** – int 32-bit

#### 4.37.567 UpdateChannelReadMessagesContents

**class** pyrogram.api.types.UpdateChannelReadMessagesContents

**ID**  
0x89893b45

**Parameters**

- **channel\_id** – int 32-bit
- **messages** – List of int 32-bit

#### 4.37.568 UpdateChannelTooLong

**class** pyrogram.api.types.UpdateChannelTooLong

**ID**

0xeb0467fb

**Parameters**

- **channel\_id** – int 32-bit
- **pts** (*optional*) – int 32-bit

#### 4.37.569 UpdateChannelWebPage

**class** pyrogram.api.types.UpdateChannelWebPage

**ID**

0x40771900

**Parameters**

- **channel\_id** – int 32-bit
- **webpage** – Either *WebPageEmpty*, *WebPagePending*, *WebPage* or *WebPageNotModified*
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

#### 4.37.570 UpdateChatDefaultBannedRights

**class** pyrogram.api.types.UpdateChatDefaultBannedRights

**ID**

0x54c01850

**Parameters**

- **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **default\_banned\_rights** – *ChatBannedRights*
- **version** – int 32-bit

### 4.37.571 UpdateChatParticipantAdd

```
class pyrogram.api.types.UpdateChatParticipantAdd
```

**ID**

0xea4b0e5c

**Parameters**

- **chat\_id** – int 32-bit
- **user\_id** – int 32-bit
- **inviter\_id** – int 32-bit
- **date** – int 32-bit
- **version** – int 32-bit

### 4.37.572 UpdateChatParticipantAdmin

```
class pyrogram.api.types.UpdateChatParticipantAdmin
```

**ID**

0xb6901959

**Parameters**

- **chat\_id** – int 32-bit
- **user\_id** – int 32-bit
- **is\_admin** – bool
- **version** – int 32-bit

### 4.37.573 UpdateChatParticipantDelete

```
class pyrogram.api.types.UpdateChatParticipantDelete
```

**ID**

0x6e5f8c22

**Parameters**

- **chat\_id** – int 32-bit
- **user\_id** – int 32-bit
- **version** – int 32-bit

#### 4.37.574 UpdateChatParticipants

**class** pyrogram.api.types.UpdateChatParticipants

**ID**

0x07761198

**Parameters** **participants** – Either *ChatParticipantsForbidden* or *ChatParticipants*

#### 4.37.575 UpdateChatPinnedMessage

**class** pyrogram.api.types.UpdateChatPinnedMessage

**ID**

0xe10db349

**Parameters**

- **chat\_id** – int 32-bit
- **id** – int 32-bit
- **version** – int 32-bit

#### 4.37.576 UpdateChatUserTyping

**class** pyrogram.api.types.UpdateChatUserTyping

**ID**

0x9a65ealf

**Parameters**

- **chat\_id** – int 32-bit
- **user\_id** – int 32-bit
- **action** – Either *SendMessageTypingAction*, *SendMessageCancelAction*, *SendMessageRecordVideoAction*, *SendMessageUploadVideoAction*, *SendMessageRecordAudioAction*, *SendMessageUploadAudioAction*, *SendMessageUploadPhotoAction*, *SendMessageUploadDocumentAction*, *SendMessageGeoLocationAction*, *SendMessageChooseContactAction*, *SendMessageGamePlayAction*, *SendMessageRecordRoundAction* or *SendMessageUploadRoundAction*

#### 4.37.577 UpdateConfig

**class** pyrogram.api.types.UpdateConfig

**ID**

0xa229dd06

No parameters required.

#### 4.37.578 UpdateContactLink

**class** pyrogram.api.types.UpdateContactLink

**ID**

0x9d2e67c5

**Parameters**

- **user\_id** – int 32-bit
- **my\_link** – Either *ContactLinkUnknown*, *ContactLinkNone*, *ContactLinkHasPhone* or *ContactLinkContact*
- **foreign\_link** – Either *ContactLinkUnknown*, *ContactLinkNone*, *ContactLinkHasPhone* or *ContactLinkContact*

#### 4.37.579 UpdateContactsReset

**class** pyrogram.api.types.UpdateContactsReset

**ID**

0x7084a7be

No parameters required.

#### 4.37.580 UpdateDcOptions

**class** pyrogram.api.types.UpdateDcOptions

**ID**

0x8e5e9873

**Parameters** **dc\_options** – List of *DcOption*

#### 4.37.581 UpdateDeleteChannelMessages

**class** pyrogram.api.types.UpdateDeleteChannelMessages

**ID**

0xc37521c9

**Parameters**

- **channel\_id** – int 32-bit
- **messages** – List of int 32-bit
- **pts** – int 32-bit
- **pts\_count** – int 32-bit



### 4.37.582 UpdateDeleteMessages

**class** pyrogram.api.types.UpdateDeleteMessages

**ID**

0xa20db0e5

**Parameters**

- **messages** – List of int 32-bit
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

### 4.37.583 UpdateDialogPinned

**class** pyrogram.api.types.UpdateDialogPinned

**ID**

0x19d27f3c

**Parameters**

- **peer** – *DialogPeer*
- **pinned**(*optional*) – bool

### 4.37.584 UpdateDialogUnreadMark

**class** pyrogram.api.types.UpdateDialogUnreadMark

**ID**

0xe16459c3

**Parameters**

- **peer** – *DialogPeer*
- **unread**(*optional*) – bool

### 4.37.585 UpdateDraftMessage

**class** pyrogram.api.types.UpdateDraftMessage

**ID**

0xee2bb969

**Parameters**

- **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **draft** – Either *DraftMessageEmpty* or *DraftMessage*

#### 4.37.586 UpdateEditChannelMessage

**class** pyrogram.api.types.UpdateEditChannelMessage

**ID**

0x1b3f4df7

**Parameters**

- **message** – Either *MessageEmpty*, *Message* or *MessageService*
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

#### 4.37.587 UpdateEditMessage

**class** pyrogram.api.types.UpdateEditMessage

**ID**

0xe40370a3

**Parameters**

- **message** – Either *MessageEmpty*, *Message* or *MessageService*
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

#### 4.37.588 UpdateEncryptedChatTyping

**class** pyrogram.api.types.UpdateEncryptedChatTyping

**ID**

0x1710f156

**Parameters** **chat\_id** – int 32-bit

#### 4.37.589 UpdateEncryptedMessagesRead

**class** pyrogram.api.types.UpdateEncryptedMessagesRead

**ID**

0x38fe25b7

**Parameters**

- **chat\_id** – int 32-bit
- **max\_date** – int 32-bit
- **date** – int 32-bit

### 4.37.590 UpdateEncryption

**class** pyrogram.api.types.UpdateEncryption

**ID**

0xb4a2e88d

**Parameters**

- **chat** – Either *EncryptedChatEmpty*, *EncryptedChatWaiting*, *EncryptedChatRequested*, *EncryptedChat* or *EncryptedChatDiscarded*
- **date** – int 32-bit

### 4.37.591 UpdateFavedStickers

**class** pyrogram.api.types.UpdateFavedStickers

**ID**

0xe511996d

No parameters required.

### 4.37.592 UpdateInlineBotCallbackQuery

**class** pyrogram.api.types.UpdateInlineBotCallbackQuery

**ID**

0xf9d27a5a

**Parameters**

- **query\_id** – int 64-bit
- **user\_id** – int 32-bit
- **msg\_id** – *InputBotInlineMessageID*
- **chat\_instance** – int 64-bit
- **data** (*optional*) – bytes
- **game\_short\_name** (*optional*) – str

### 4.37.593 UpdateLangPack

**class** pyrogram.api.types.UpdateLangPack

**ID**

0x56022f4d

**Parameters** **difference** – *LangPackDifference*

#### 4.37.594 UpdateLangPackTooLong

**class** pyrogram.api.types.UpdateLangPackTooLong

**ID**

0x46560264

**Parameters** `lang_code` – str

#### 4.37.595 UpdateMessageID

**class** pyrogram.api.types.UpdateMessageID

**ID**

0x4e90bfd6

**Parameters**

- `id` – int 32-bit
- `random_id` – int 64-bit

#### 4.37.596 UpdateMessagePoll

**class** pyrogram.api.types.UpdateMessagePoll

**ID**

0xacal657b

**Parameters**

- `poll_id` – int 64-bit
- `results` – *PollResults*
- `poll` (*optional*) – *Poll*

#### 4.37.597 UpdateNewChannelMessage

**class** pyrogram.api.types.UpdateNewChannelMessage

**ID**

0x62ba04d9

**Parameters**

- `message` – Either *MessageEmpty*, *Message* or *MessageService*
- `pts` – int 32-bit
- `pts_count` – int 32-bit

### 4.37.598 UpdateNewEncryptedMessage

**class** pyrogram.api.types.UpdateNewEncryptedMessage

**ID**  
0x12bcbd9a

**Parameters**

- **message** – Either *EncryptedMessage* or *EncryptedMessageService*
- **qts** – int 32-bit

### 4.37.599 UpdateNewMessage

**class** pyrogram.api.types.UpdateNewMessage

**ID**  
0x1f2b0afd

**Parameters**

- **message** – Either *MessageEmpty*, *Message* or *MessageService*
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

### 4.37.600 UpdateNewStickerSet

**class** pyrogram.api.types.UpdateNewStickerSet

**ID**  
0x688a30aa

**Parameters** **stickerset** – *messages.StickerSet*

### 4.37.601 UpdateNotifySettings

**class** pyrogram.api.types.UpdateNotifySettings

**ID**  
0xbec268ef

**Parameters**

- **peer** – Either *NotifyPeer*, *NotifyUsers*, *NotifyChats* or *NotifyBroadcasts*
- **notify\_settings** – *PeerNotifySettings*

### 4.37.602 UpdatePhoneCall

**class** pyrogram.api.types.UpdatePhoneCall

**ID**

0xab0f6b1e

**Parameters** **phone\_call** – Either *PhoneCallEmpty*, *PhoneCallWaiting*,  
*PhoneCallRequested*, *PhoneCallAccepted*, *PhoneCall* or  
*PhoneCallDiscarded*

### 4.37.603 UpdatePinnedDialogs

**class** pyrogram.api.types.UpdatePinnedDialogs

**ID**

0xea4cb65b

**Parameters** **order** (*optional*) – List of *DialogPeer*

### 4.37.604 UpdatePrivacy

**class** pyrogram.api.types.UpdatePrivacy

**ID**

0xee3b272a

**Parameters**

- **key** – Either *PrivacyKeyStatusTimestamp*, *PrivacyKeyChatInvite*,  
*PrivacyKeyPhoneCall*, *PrivacyKeyPhoneP2P*, *PrivacyKeyForwards* or  
*PrivacyKeyProfilePhoto*
- **rules** – List of either *PrivacyValueAllowContacts*,  
*PrivacyValueAllowAll*, *PrivacyValueAllowUsers*,  
*PrivacyValueDisallowContacts*, *PrivacyValueDisallowAll* or  
*PrivacyValueDisallowUsers*

### 4.37.605 UpdatePtsChanged

**class** pyrogram.api.types.UpdatePtsChanged

**ID**

0x3354678f

No parameters required.

#### 4.37.606 UpdateReadChannelInbox

**class** pyrogram.api.types.UpdateReadChannelInbox

**ID**  
0x4214f37f

**Parameters**

- **channel\_id** – int 32-bit
- **max\_id** – int 32-bit

#### 4.37.607 UpdateReadChannelOutbox

**class** pyrogram.api.types.UpdateReadChannelOutbox

**ID**  
0x25d6c9c7

**Parameters**

- **channel\_id** – int 32-bit
- **max\_id** – int 32-bit

#### 4.37.608 UpdateReadFeaturedStickers

**class** pyrogram.api.types.UpdateReadFeaturedStickers

**ID**  
0x571d2742

No parameters required.

#### 4.37.609 UpdateReadHistoryInbox

**class** pyrogram.api.types.UpdateReadHistoryInbox

**ID**  
0x9961fd5c

**Parameters**

- **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **max\_id** – int 32-bit
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

#### 4.37.610 UpdateReadHistoryOutbox

**class** pyrogram.api.types.UpdateReadHistoryOutbox

**ID**

0x2f2f21bf

**Parameters**

- **peer** – Either *PeerUser*, *PeerChat* or *PeerChannel*
- **max\_id** – int 32-bit
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

#### 4.37.611 UpdateReadMessagesContents

**class** pyrogram.api.types.UpdateReadMessagesContents

**ID**

0x68c13933

**Parameters**

- **messages** – List of int 32-bit
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

#### 4.37.612 UpdateRecentStickers

**class** pyrogram.api.types.UpdateRecentStickers

**ID**

0x9a422c20

No parameters required.

#### 4.37.613 UpdateSavedGifs

**class** pyrogram.api.types.UpdateSavedGifs

**ID**

0x9375341e

No parameters required.



### 4.37.614 UpdateServiceNotification

**class** pyrogram.api.types.UpdateServiceNotification

**ID**

0xebe46819

#### Parameters

- **type** – str
- **message** – str
- **media** – Either *MessageMediaEmpty*, *MessageMediaPhoto*, *MessageMediaGeo*, *MessageMediaContact*, *MessageMediaUnsupported*, *MessageMediaDocument*, *MessageMediaWebPage*, *MessageMediaVenue*, *MessageMediaGame*, *MessageMediaInvoice*, *MessageMediaGeoLive* or *MessageMediaPoll*
- **entities** – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*
- **popup** (optional) – bool
- **inbox\_date** (optional) – int 32-bit

### 4.37.615 UpdateShort

**class** pyrogram.api.types.UpdateShort

**ID**

0x78d4dec1

#### Parameters

- **update** – Either *UpdateNewMessage*, *UpdateMessageID*, *UpdateDeleteMessages*, *UpdateUserTyping*, *UpdateChatUserTyping*, *UpdateChatParticipants*, *UpdateUserStatus*, *UpdateUserName*, *UpdateUserPhoto*, *UpdateContactLink*, *UpdateNewEncryptedMessage*, *UpdateEncryptedChatTyping*, *UpdateEncryption*, *UpdateEncryptedMessagesRead*, *UpdateChatParticipantAdd*, *UpdateChatParticipantDelete*, *UpdateDcOptions*, *UpdateUserBlocked*, *UpdateNotifySettings*, *UpdateServiceNotification*, *UpdatePrivacy*, *UpdateUserPhone*, *UpdateReadHistoryInbox*, *UpdateReadHistoryOutbox*, *UpdateWebPage*, *UpdateReadMessagesContents*, *UpdateChannelTooLong*, *UpdateChannel*, *UpdateNewChannelMessages*, *UpdateReadChannelInbox*, *UpdateDeleteChannelMessages*, *UpdateChannelMessageViews*, *UpdateChatParticipantAdmin*, *UpdateNewStickerSet*, *UpdateStickerSetsOrder*, *UpdateStickerSets*,

```
UpdateSavedGifs, UpdateBotInlineQuery, UpdateBotInlineSend,
UpdateEditChannelMessage, UpdateChannelPinnedMessage,
UpdateBotCallbackQuery, UpdateEditMessage,
UpdateInlineBotCallbackQuery, UpdateReadChannelOutbox,
UpdateDraftMessage, UpdateReadFeaturedStickers,
UpdateRecentStickers, UpdateConfig, UpdatePtsChanged,
UpdateChannelWebPage, UpdateDialogPinned, UpdatePinnedDialogs,
UpdateBotWebhookJSON, UpdateBotWebhookJSONQuery,
UpdateBotShippingQuery, UpdateBotPrecheckoutQuery,
UpdatePhoneCall, UpdateLangPackTooLong, UpdateLangPack,
UpdateFavedStickers, UpdateChannelReadMessagesContents,
UpdateContactsReset, UpdateChannelAvailableMessages,
UpdateDialogUnreadMark, UpdateUserPinnedMessage,
UpdateChatPinnedMessage, UpdateMessagePoll or
UpdateChatDefaultBannedRights
```

- **date** – int 32-bit

#### See also:

This object can be returned by `account.GetNotifyExceptions`, `messages.SendMessage`, `messages.SendMedia`, `messages.ForwardMessages`, `messages.EditChatTitle`, `messages.EditChatPhoto`, `messages.AddChatUser`, `messages.DeleteChatUser`, `messages.CreateChat`, `messages.ImportChatInvite`, `messages.StartBot`, `messages.MigrateChat`, `messages.SendInlineBotResult`, `messages.EditMessage`, `messages.GetAllDrafts`, `messages.SetGameScore`, `messages.SendScreenshotNotification`, `messages.SendMultiMedia`, `messages.UpdatePinnedMessage`, `messages.SendVote`, `messages.GetPollResults`, `messages.EditChatDefaultBannedRights`, `help.GetAppChangelog`, `channels.CreateChannel`, `channels.EditAdmin`, `channels.EditTitle`, `channels.EditPhoto`, `channels.JoinChannel`, `channels.LeaveChannel`, `channels.InviteToChannel`, `channels.DeleteChannel`, `channels.ToggleSignatures`, `channels.EditBanned`, `channels.TogglePreHistoryHidden`, `phone.DiscardCall` and `phone.SetCallRating`.

### 4.37.616 UpdateShortChatMessage

**class** pyrogram.api.types.UpdateShortChatMessage

#### ID

0x16812688

#### Parameters

- **id** – int 32-bit
- **from\_id** – int 32-bit
- **chat\_id** – int 32-bit
- **message** – str
- **pts** – int 32-bit
- **pts\_count** – int 32-bit
- **date** – int 32-bit
- **out** (*optional*) – bool

- **mentioned**(*optional*) – bool
- **media\_unread**(*optional*) – bool
- **silent**(*optional*) – bool
- **fwd\_from**(*optional*) – *MessageFwdHeader*
- **via\_bot\_id**(*optional*) – int 32-bit
- **reply\_to\_msg\_id**(*optional*) – int 32-bit
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

**See also:**

This object can be returned by *account.GetNotifyExceptions*, *messages.SendMessage*, *messages.SendMedia*, *messages.ForwardMessages*, *messages.EditChatTitle*, *messages.EditChatPhoto*, *messages.AddChatUser*, *messages.DeleteChatUser*, *messages.CreateChat*, *messages.ImportChatInvite*, *messages.StartBot*, *messages.MigrateChat*, *messages.SendInlineBotResult*, *messages.EditMessage*, *messages.GetAllDrafts*, *messages.SetGameScore*, *messages.SendScreenshotNotification*, *messages.SendMultiMedia*, *messages.UpdatePinnedMessage*, *messages.SendVote*, *messages.GetPollResults*, *messages.EditChatDefaultBannedRights*, *help.GetAppChangelog*, *channels.CreateChannel*, *channels.EditAdmin*, *channels.EditTitle*, *channels.EditPhoto*, *channels.JoinChannel*, *channels.LeaveChannel*, *channels.InviteToChannel*, *channels.DeleteChannel*, *channels.ToggleSignatures*, *channels.EditBanned*, *channels.TogglePreHistoryHidden*, *phone.DiscardCall* and *phone.SetCallRating*.

### 4.37.617 UpdateShortMessage

**class** pyrogram.api.types.UpdateShortMessage

**ID**

0x914fbf11

**Parameters**

- **id** – int 32-bit
- **user\_id** – int 32-bit
- **message** – str
- **pts** – int 32-bit
- **pts\_count** – int 32-bit
- **date** – int 32-bit
- **out** (*optional*) – bool
- **mentioned** (*optional*) – bool

- **media\_unread**(*optional*) – bool
- **silent**(*optional*) – bool
- **fwd\_from**(*optional*) – *MessageFwdHeader*
- **via\_bot\_id**(*optional*) – int 32-bit
- **reply\_to\_msg\_id**(*optional*) – int 32-bit
- **entities** (*optional*) – List of either *MessageEntityUnknown*, *MessageEntityMention*, *MessageEntityHashtag*, *MessageEntityBotCommand*, *MessageEntityUrl*, *MessageEntityEmail*, *MessageEntityBold*, *MessageEntityItalic*, *MessageEntityCode*, *MessageEntityPre*, *MessageEntityTextUrl*, *MessageEntityMentionName*, *InputMessageEntityMentionName*, *MessageEntityPhone* or *MessageEntityCashtag*

**See also:**

This object can be returned by *account.GetNotifyExceptions*, *messages.SendMessage*, *messages.SendMedia*, *messages.ForwardMessages*, *messages.EditChatTitle*, *messages.EditChatPhoto*, *messages.AddChatUser*, *messages.DeleteChatUser*, *messages.CreateChat*, *messages.ImportChatInvite*, *messages.StartBot*, *messages.MigrateChat*, *messages.SendInlineBotResult*, *messages.EditMessage*, *messages.GetAllDrafts*, *messages.SetGameScore*, *messages.SendScreenshotNotification*, *messages.SendMultiMedia*, *messages.UpdatePinnedMessage*, *messages.SendVote*, *messages.GetPollResults*, *messages.EditChatDefaultBannedRights*, *help.GetAppChangelog*, *channels.CreateChannel*, *channels.EditAdmin*, *channels.EditTitle*, *channels.EditPhoto*, *channels.JoinChannel*, *channels.LeaveChannel*, *channels.InviteToChannel*, *channels.DeleteChannel*, *channels.ToggleSignatures*, *channels.EditBanned*, *channels.TogglePreHistoryHidden*, *phone.DiscardCall* and *phone.SetCallRating*.

## 4.37.618 UpdateShortSentMessage

**class** pyrogram.api.types.UpdateShortSentMessage

**ID**

0x11f1331c

**Parameters**

- **id** – int 32-bit
- **pts** – int 32-bit
- **pts\_count** – int 32-bit
- **date** – int 32-bit
- **out** (*optional*) – bool
- **media** (*optional*) – Either *MessageMediaEmpty*, *MessageMediaPhoto*, *MessageMediaGeo*, *MessageMediaContact*, *MessageMediaUnsupported*, *MessageMediaDocument*, *MessageMediaWebPage*, *MessageMediaVenue*, *MessageMediaGame*, *MessageMediaInvoice*, *MessageMediaGeoLive* or *MessageMediaPoll*

- **entities** (optional) – List of either `MessageEntityUnknown`, `MessageEntityMention`, `MessageEntityHashtag`, `MessageEntityBotCommand`, `MessageEntityUrl`, `MessageEntityEmail`, `MessageEntityBold`, `MessageEntityItalic`, `MessageEntityCode`, `MessageEntityPre`, `MessageEntityTextUrl`, `MessageEntityMentionName`, `InputMessageEntityMentionName`, `MessageEntityPhone` or `MessageEntityCashtag`

#### See also:

This object can be returned by `account.GetNotifyExceptions`, `messages.SendMessage`, `messages.SendMedia`, `messages.ForwardMessages`, `messages.EditChatTitle`, `messages.EditChatPhoto`, `messages.AddChatUser`, `messages.DeleteChatUser`, `messages.CreateChat`, `messages.ImportChatInvite`, `messages.StartBot`, `messages.MigrateChat`, `messages.SendInlineBotResult`, `messages.EditMessage`, `messages.GetAllDrafts`, `messages.SetGameScore`, `messages.SendScreenshotNotification`, `messages.SendMultiMedia`, `messages.UpdatePinnedMessage`, `messages.SendVote`, `messages.GetPollResults`, `messages.EditChatDefaultBannedRights`, `help.GetAppChangelog`, `channels.CreateChannel`, `channels.EditAdmin`, `channels.EditTitle`, `channels.EditPhoto`, `channels.JoinChannel`, `channels.LeaveChannel`, `channels.InviteToChannel`, `channels.DeleteChannel`, `channels.ToggleSignatures`, `channels.EditBanned`, `channels.TogglePreHistoryHidden`, `phone.DiscardCall` and `phone.SetCallRating`.

### 4.37.619 UpdateStickerSets

**class** pyrogram.api.types.UpdateStickerSets

#### ID

0x43ae3dec

No parameters required.

### 4.37.620 UpdateStickerSetsOrder

**class** pyrogram.api.types.UpdateStickerSetsOrder

#### ID

0x0bb2d201

#### Parameters

- **order** – List of int 64-bit
- **masks** (optional) – bool

### 4.37.621 UpdateUserBlocked

**class** pyrogram.api.types.UpdateUserBlocked

#### ID

0x80ece81a

**Parameters**

- **user\_id** – int 32-bit
- **blocked** – bool

### 4.37.622 UpdateUserName

```
class pyrogram.api.types.UpdateUserName
```

**ID**

0xa7332b73

**Parameters**

- **user\_id** – int 32-bit
- **first\_name** – str
- **last\_name** – str
- **username** – str

### 4.37.623 UpdateUserPhone

```
class pyrogram.api.types.UpdateUserPhone
```

**ID**

0x12b9417b

**Parameters**

- **user\_id** – int 32-bit
- **phone** – str

### 4.37.624 UpdateUserPhoto

```
class pyrogram.api.types.UpdateUserPhoto
```

**ID**

0x95313b0c

**Parameters**

- **user\_id** – int 32-bit
- **date** – int 32-bit
- **photo** – Either *UserProfilePhotoEmpty* or *UserProfilePhoto*
- **previous** – bool

#### 4.37.625 updateUserPinnedMessage

**class** pyrogram.api.types.UpdateUserPinnedMessage

**ID**

0x4c43da18

**Parameters**

- **user\_id** – int 32-bit
- **id** – int 32-bit

#### 4.37.626 updateUserStatus

**class** pyrogram.api.types.UpdateUserStatus

**ID**

0x1bfbd823

**Parameters**

- **user\_id** – int 32-bit
- **status** – Either *UserStatusEmpty*, *UserStatusOnline*, *UserStatusOffline*, *UserStatusRecently*, *UserStatusLastWeek* or *UserStatusLastMonth*

#### 4.37.627 updateUserTyping

**class** pyrogram.api.types.UpdateUserTyping

**ID**

0x5c486927

**Parameters**

- **user\_id** – int 32-bit
- **action** – Either *SendMessageTypingAction*, *SendMessageCancelAction*, *SendMessageRecordVideoAction*, *SendMessageUploadVideoAction*, *SendMessageRecordAudioAction*, *SendMessageUploadAudioAction*, *SendMessageUploadPhotoAction*, *SendMessageUploadDocumentAction*, *SendMessageGeoLocationAction*, *SendMessageChooseContactAction*, *SendMessageGamePlayAction*, *SendMessageRecordRoundAction* or *SendMessageUploadRoundAction*

#### 4.37.628 UpdateWebPage

**class** pyrogram.api.types.UpdateWebPage

**ID**

0x7f891213

**Parameters**

- **webpage** – Either *WebPageEmpty*, *WebPagePending*, *WebPage* or *WebPageNotModified*
- **pts** – int 32-bit
- **pts\_count** – int 32-bit

## 4.37.629 UpdatesCombined

**class** pyrogram.api.types.**UpdatesCombined****ID**

0x725b04c3

**Parameters**

- **updates** – List of either *UpdateNewMessage*, *UpdateMessageID*, *UpdateDeleteMessages*, *UpdateUserTyping*, *UpdateChatUserTyping*, *UpdateChatParticipants*, *UpdateUserStatus*, *UpdateUserName*, *UpdateUserPhoto*, *UpdateContactLink*, *UpdateNewEncryptedMessage*, *UpdateEncryptedChatTyping*, *UpdateEncryption*, *UpdateEncryptedMessagesRead*, *UpdateChatParticipantAdd*, *UpdateChatParticipantDelete*, *UpdateDcOptions*, *UpdateUserBlocked*, *UpdateNotifySettings*, *UpdateServiceNotification*, *UpdatePrivacy*, *UpdateUserPhone*, *UpdateReadHistoryInbox*, *UpdateReadHistoryOutbox*, *UpdateWebPage*, *UpdateReadMessagesContents*, *UpdateChannelTooLong*, *UpdateChannel*, *UpdateNewChannelMessage*, *UpdateReadChannelInbox*, *UpdateDeleteChannelMessages*, *UpdateChannelMessageViews*, *UpdateChatParticipantAdmin*, *UpdateNewStickerSet*, *UpdateStickerSetsOrder*, *UpdateStickerSets*, *UpdateSavedGifs*, *UpdateBotInlineQuery*, *UpdateBotInlineSend*, *UpdateEditChannelMessage*, *UpdateChannelPinnedMessage*, *UpdateBotCallbackQuery*, *UpdateEditMessage*, *UpdateInlineBotCallbackQuery*, *UpdateReadChannelOutbox*, *UpdateDraftMessage*, *UpdateReadFeaturedStickers*, *UpdateRecentStickers*, *UpdateConfig*, *UpdatePtsChanged*, *UpdateChannelWebPage*, *UpdateDialogPinned*, *UpdatePinnedDialogs*, *UpdateBotWebhookJSON*, *UpdateBotWebhookJSONQuery*, *UpdateBotShippingQuery*, *UpdateBotPrecheckoutQuery*, *UpdatePhoneCall*, *UpdateLangPackTooLong*, *UpdateLangPack*, *UpdateFavedStickers*, *UpdateChannelReadMessagesContents*, *UpdateContactsReset*, *UpdateChannelAvailableMessages*, *UpdateDialogUnreadMark*, *UpdateUserPinnedMessage*, *UpdateChatPinnedMessage*, *UpdateMessagePoll* or *UpdateChatDefaultBannedRights*
- **users** – List of either *UserEmpty* or *User*
- **chats** – List of either *ChatEmpty*, *Chat*, *ChatForbidden*, *Channel* or *ChannelForbidden*



- **date** – int 32-bit
- **seq\_start** – int 32-bit
- **seq** – int 32-bit

**See also:**

This object can be returned by `account.GetNotifyExceptions`, `messages.SendMessage`, `messages.SendMedia`, `messages.ForwardMessages`, `messages.EditChatTitle`, `messages.EditChatPhoto`, `messages.AddChatUser`, `messages.DeleteChatUser`, `messages.CreateChat`, `messages.ImportChatInvite`, `messages.StartBot`, `messages.MigrateChat`, `messages.SendInlineBotResult`, `messages.EditMessage`, `messages.GetAllDrafts`, `messages.SetGameScore`, `messages.SendScreenshotNotification`, `messages.SendMultiMedia`, `messages.UpdatePinnedMessage`, `messages.SendVote`, `messages.GetPollResults`, `messages.EditChatDefaultBannedRights`, `help.GetAppChangelog`, `channels.CreateChannel`, `channels.EditAdmin`, `channels.EditTitle`, `channels.EditPhoto`, `channels.JoinChannel`, `channels.LeaveChannel`, `channels.InviteToChannel`, `channels.DeleteChannel`, `channels.ToggleSignatures`, `channels.EditBanned`, `channels.TogglePreHistoryHidden`, `phone.DiscardCall` and `phone.SetCallRating`.

**4.37.630 UpdatesTooLong**

**class** pyrogram.api.types.**UpdatesTooLong**

**ID**

0xe317af7e

No parameters required.

**See also:**

This object can be returned by `account.GetNotifyExceptions`, `messages.SendMessage`, `messages.SendMedia`, `messages.ForwardMessages`, `messages.EditChatTitle`, `messages.EditChatPhoto`, `messages.AddChatUser`, `messages.DeleteChatUser`, `messages.CreateChat`, `messages.ImportChatInvite`, `messages.StartBot`, `messages.MigrateChat`, `messages.SendInlineBotResult`, `messages.EditMessage`, `messages.GetAllDrafts`, `messages.SetGameScore`, `messages.SendScreenshotNotification`, `messages.SendMultiMedia`, `messages.UpdatePinnedMessage`, `messages.SendVote`, `messages.GetPollResults`, `messages.EditChatDefaultBannedRights`, `help.GetAppChangelog`, `channels.CreateChannel`, `channels.EditAdmin`, `channels.EditTitle`, `channels.EditPhoto`, `channels.JoinChannel`, `channels.LeaveChannel`, `channels.InviteToChannel`, `channels.DeleteChannel`, `channels.ToggleSignatures`, `channels.EditBanned`, `channels.TogglePreHistoryHidden`, `phone.DiscardCall` and `phone.SetCallRating`.

**4.37.631 User**

**class** pyrogram.api.types.**User**

**ID**

0x2e13f4c3

**Parameters**

- **id** – int 32-bit
- **is\_self** (*optional*) – bool
- **contact** (*optional*) – bool
- **mutual\_contact** (*optional*) – bool
- **deleted** (*optional*) – bool
- **bot** (*optional*) – bool
- **bot\_chat\_history** (*optional*) – bool
- **bot\_nochats** (*optional*) – bool
- **verified** (*optional*) – bool
- **restricted** (*optional*) – bool
- **min** (*optional*) – bool
- **bot\_inline\_geo** (*optional*) – bool
- **support** (*optional*) – bool
- **access\_hash** (*optional*) – int 64-bit
- **first\_name** (*optional*) – str
- **last\_name** (*optional*) – str
- **username** (*optional*) – str
- **phone** (*optional*) – str
- **photo** (*optional*) – Either *UserProfilePhotoEmpty* or *UserProfilePhoto*
- **status** (*optional*) – Either *UserStatusEmpty*, *UserStatusOnline*, *UserStatusOffline*, *UserStatusRecently*, *UserStatusLastWeek* or *UserStatusLastMonth*
- **bot\_info\_version** (*optional*) – int 32-bit
- **restriction\_reason** (*optional*) – str
- **bot\_inline\_placeholder** (*optional*) – str
- **lang\_code** (*optional*) – str

**See also:**

This object can be returned by *account.UpdateProfile*, *account.UpdateUsername*, *account.ChangePhone* and *users.GetUsers*.

### 4.37.632 UserEmpty

**class** pyrogram.api.types.**UserEmpty**

**ID**

0x200250ba

**Parameters** **id** – int 32-bit

**See also:**

This object can be returned by `account.UpdateProfile`, `account.UpdateUsername`, `account.ChangePhone` and `users.GetUsers`.

**4.37.633 UserFull**

```
class pyrogram.api.types.UserFull
```

**ID**

0x8ea4a881

**Parameters**

- **user** – Either *UserEmpty* or *User*
- **link** – *contacts.Link*
- **notify\_settings** – *PeerNotifySettings*
- **common\_chats\_count** – int 32-bit
- **blocked**(*optional*) – bool
- **phone\_calls\_available**(*optional*) – bool
- **phone\_calls\_private**(*optional*) – bool
- **can\_pin\_message**(*optional*) – bool
- **about**(*optional*) – str
- **profile\_photo**(*optional*) – Either *PhotoEmpty* or *Photo*
- **bot\_info**(*optional*) – *BotInfo*
- **pinned\_msg\_id**(*optional*) – int 32-bit

**See also:**

This object can be returned by `users.GetFullUser`.

**4.37.634 UserProfilePhoto**

```
class pyrogram.api.types.UserProfilePhoto
```

**ID**

0xd559d8c8

**Parameters**

- **photo\_id** – int 64-bit
- **photo\_small** – Either *FileLocationUnavailable* or *FileLocation*
- **photo\_big** – Either *FileLocationUnavailable* or *FileLocation*

**See also:**

This object can be returned by `photos.UpdateProfilePhoto`.

#### 4.37.635 UserProfilePhotoEmpty

```
class pyrogram.api.types.UserProfilePhotoEmpty
```

**ID**

0x4f11bae1

No parameters required.

**See also:**

This object can be returned by *photos.UpdateProfilePhoto*.

#### 4.37.636 UserStatusEmpty

```
class pyrogram.api.types.UserStatusEmpty
```

**ID**

0x09d05049

No parameters required.

#### 4.37.637 UserStatusLastMonth

```
class pyrogram.api.types.UserStatusLastMonth
```

**ID**

0x77ebc742

No parameters required.

#### 4.37.638 UserStatusLastWeek

```
class pyrogram.api.types.UserStatusLastWeek
```

**ID**

0x07bf09fc

No parameters required.

#### 4.37.639 UserStatusOffline

```
class pyrogram.api.types.UserStatusOffline
```

**ID**

0x008c703f

**Parameters** `was_online` – int 32-bit

#### 4.37.640 UserStatusOnline

**class** pyrogram.api.types.UserStatusOnline

**ID**

0xedb93949

**Parameters** **expires** – int 32-bit

#### 4.37.641 UserStatusRecently

**class** pyrogram.api.types.UserStatusRecently

**ID**

0xe26f42f1

No parameters required.

#### 4.37.642 Wallpaper

**class** pyrogram.api.types.WallPaper

**ID**

0xa437c3ed

**Parameters**

- **id** – int 64-bit
- **access\_hash** – int 64-bit
- **slug** – str
- **document** – Either *DocumentEmpty* or *Document*
- **creator** (*optional*) – bool
- **default** (*optional*) – bool
- **pattern** (*optional*) – bool
- **dark** (*optional*) – bool
- **settings** (*optional*) – *WallPaperSettings*

**See also:**

This object can be returned by *account.GetWallPaper* and *account.UploadWallPaper*.

#### 4.37.643 WallpaperSettings

**class** pyrogram.api.types.WallPaperSettings

**ID**

0xa12f40b8

**Parameters**

- **blur** (*optional*) – bool
- **motion** (*optional*) – bool
- **background\_color** (*optional*) – int 32-bit
- **intensity** (*optional*) – int 32-bit

#### 4.37.644 WebAuthorization

```
class pyrogram.api.types.WebAuthorization
```

**ID**

0xcac943f2

**Parameters**

- **hash** – int 64-bit
- **bot\_id** – int 32-bit
- **domain** – str
- **browser** – str
- **platform** – str
- **date\_created** – int 32-bit
- **date\_active** – int 32-bit
- **ip** – str
- **region** – str

#### 4.37.645 WebDocument

```
class pyrogram.api.types.WebDocument
```

**ID**

0x1c570ed1

**Parameters**

- **url** – str
- **access\_hash** – int 64-bit
- **size** – int 32-bit
- **mime\_type** – str
- **attributes** – List of either *DocumentAttributeImageSize*,  
*DocumentAttributeAnimated*, *DocumentAttributeSticker*,  
*DocumentAttributeVideo*, *DocumentAttributeAudio*,  
*DocumentAttributeFilename* or *DocumentAttributeHasStickers*

### 4.37.646 WebDocumentNoProxy

**class** pyrogram.api.types.WebDocumentNoProxy

**ID**

0xf9c8bcc6

**Parameters**

- **url** – str
- **size** – int 32-bit
- **mime\_type** – str
- **attributes** – List of either *DocumentAttributeImageSize*, *DocumentAttributeAnimated*, *DocumentAttributeSticker*, *DocumentAttributeVideo*, *DocumentAttributeAudio*, *DocumentAttributeFilename* or *DocumentAttributeHasStickers*

### 4.37.647 WebPage

**class** pyrogram.api.types.WebPage

**ID**

0x5f07b4bc

**Parameters**

- **id** – int 64-bit
- **url** – str
- **display\_url** – str
- **hash** – int 32-bit
- **type** (*optional*) – str
- **site\_name** (*optional*) – str
- **title** (*optional*) – str
- **description** (*optional*) – str
- **photo** (*optional*) – Either *PhotoEmpty* or *Photo*
- **embed\_url** (*optional*) – str
- **embed\_type** (*optional*) – str
- **embed\_width** (*optional*) – int 32-bit
- **embed\_height** (*optional*) – int 32-bit
- **duration** (*optional*) – int 32-bit
- **author** (*optional*) – str
- **document** (*optional*) – Either *DocumentEmpty* or *Document*
- **cached\_page** (*optional*) – *Page*

**See also:**

This object can be returned by `messages.GetWebPage`.

### 4.37.648 WebPageEmpty

`class pyrogram.api.types.WebPageEmpty`

**ID**

0xeb1477e8

**Parameters** `id` – int 64-bit

**See also:**

This object can be returned by `messages.GetWebPage`.

### 4.37.649 WebPageNotModified

`class pyrogram.api.types.WebPageNotModified`

**ID**

0x85849473

No parameters required.

**See also:**

This object can be returned by `messages.GetWebPage`.

### 4.37.650 WebPagePending

`class pyrogram.api.types.WebPagePending`

**ID**

0xc586da1c

**Parameters**

- `id` – int 64-bit
- `date` – int 32-bit

**See also:**

This object can be returned by `messages.GetWebPage`.



### p

- `pyrogram.api.functions`, 143
  - `pyrogram.api.functions.account`, 143
  - `pyrogram.api.functions.auth`, 158
  - `pyrogram.api.functions.bots`, 162
  - `pyrogram.api.functions.channels`, 162
  - `pyrogram.api.functions.contacts`, 171
  - `pyrogram.api.functions.contest`, 175
  - `pyrogram.api.functions.help`, 176
  - `pyrogram.api.functions.langpack`, 180
  - `pyrogram.api.functions.messages`, 182
  - `pyrogram.api.functions.payments`, 217
  - `pyrogram.api.functions.phone`, 219
  - `pyrogram.api.functions.photos`, 221
  - `pyrogram.api.functions.stickers`, 222
  - `pyrogram.api.functions.updates`, 223
  - `pyrogram.api.functions.upload`, 224
  - `pyrogram.api.functions.users`, 227
- `pyrogram.api.types`, 232
  - `pyrogram.api.types.account`, 232
  - `pyrogram.api.types.auth`, 237
  - `pyrogram.api.types.channels`, 240
  - `pyrogram.api.types.contacts`, 241
  - `pyrogram.api.types.help`, 245
  - `pyrogram.api.types.messages`, 251
  - `pyrogram.api.types.payments`, 264
  - `pyrogram.api.types.phone`, 266
  - `pyrogram.api.types.photos`, 267
  - `pyrogram.api.types.storage`, 268
  - `pyrogram.api.types.updates`, 270
  - `pyrogram.api.types.upload`, 275



## A

- AcceptAuthorization (class in pyrogram.api.functions.account), 143
- AcceptCall (class in pyrogram.api.functions.phone), 219
- AcceptEncryption (class in pyrogram.api.functions.messages), 182
- AcceptTermsOfService (class in pyrogram.api.functions.help), 176
- AccessPointRule (class in pyrogram.api.types), 276
- AccountDaysTTL (class in pyrogram.api.types), 277
- add\_contacts() (pyrogram.Client method), 50
- add\_handler() (pyrogram.Client method), 23
- AddChatUser (class in pyrogram.api.functions.messages), 182
- AddStickerToSet (class in pyrogram.api.functions.stickers), 222
- AdminLogResults (class in pyrogram.api.types.channels), 240
- AffectedHistory (class in pyrogram.api.types.messages), 251
- AffectedMessages (class in pyrogram.api.types.messages), 251
- AllStickers (class in pyrogram.api.types.messages), 252
- AllStickersNotModified (class in pyrogram.api.types.messages), 252
- Animation (class in pyrogram), 65
- animation (pyrogram.Filters attribute), 104
- answer() (pyrogram.CallbackQuery method), 96
- answer() (pyrogram.InlineQuery method), 97
- answer\_callback\_query() (pyrogram.Client method), 52
- answer\_inline\_query() (pyrogram.Client method), 53
- AnswerWebhookJSONQuery (class in pyrogram.api.functions.bots), 162
- API, 141
- API key, 141
- AppUpdate (class in pyrogram.api.types.help), 245
- ArchivedStickers (class in pyrogram.api.types.messages), 252
- Audio (class in pyrogram), 65
- audio (pyrogram.Filters attribute), 104
- Authorization (class in pyrogram.api.types), 277
- Authorization (class in pyrogram.api.types.auth), 237
- AuthorizationForm (class in pyrogram.api.types.account), 233
- Authorizations (class in pyrogram.api.types.account), 233
- AutoDownloadSettings (class in pyrogram.api.types), 278
- AutoDownloadSettings (class in pyrogram.api.types.account), 233

## B

- BadMsgNotification (class in pyrogram.api.types), 278
- BadServerSalt (class in pyrogram.api.types), 278
- BindAuthKeyInner (class in pyrogram.api.types), 279
- BindTempAuthKey (class in pyrogram.api.functions.auth), 158
- Block (class in pyrogram.api.functions.contacts), 171
- Blocked (class in pyrogram.api.types.contacts), 241
- BlockedSlice (class in pyrogram.api.types.contacts), 242
- bot (pyrogram.Filters attribute), 103
- Bot API, 141
- BotCallbackAnswer (class in pyrogram.api.types.messages), 252
- BotCommand (class in pyrogram.api.types), 279
- BotInfo (class in pyrogram.api.types), 279
- BotInlineMediaResult (class in pyrogram.api.types), 279
- BotInlineMessageMediaAuto (class in pyrogram.api.types), 280

BotInlineMessageMediaContact (class in pyrogram.api.types), 280  
BotInlineMessageMediaGeo (class in pyrogram.api.types), 281  
BotInlineMessageMediaVenue (class in pyrogram.api.types), 281  
BotInlineMessageText (class in pyrogram.api.types), 281  
BotInlineResult (class in pyrogram.api.types), 282  
BotResults (class in pyrogram.api.types.messages), 253

## C

Callback, 141  
CallbackGame (class in pyrogram), 70  
CallbackQuery (class in pyrogram), 70  
CallbackQueryHandler (class in pyrogram), 99  
CancelCode (class in pyrogram.api.functions.auth), 158  
CancelPasswordEmail (class in pyrogram.api.functions.account), 144  
caption (pyrogram.Filters attribute), 104  
CdnConfig (class in pyrogram.api.types), 282  
CdnFile (class in pyrogram.api.types.upload), 275  
CdnFileReuploadNeeded (class in pyrogram.api.types.upload), 275  
CdnPublicKey (class in pyrogram.api.types), 283  
change\_cloud\_password() (pyrogram.Client method), 51  
ChangePhone (class in pyrogram.api.functions.account), 144  
ChangeStickerPosition (class in pyrogram.api.functions.stickers), 223  
Channel (class in pyrogram.api.types), 283  
channel (pyrogram.Filters attribute), 105  
channel\_chat\_created (pyrogram.Filters attribute), 105  
ChannelAdminLogEvent (class in pyrogram.api.types), 284  
ChannelAdminLogEventActionChangeAbout (class in pyrogram.api.types), 284  
ChannelAdminLogEventActionChangePhoto (class in pyrogram.api.types), 284  
ChannelAdminLogEventActionChangeStickerSet (class in pyrogram.api.types), 285  
ChannelAdminLogEventActionChangeTitle (class in pyrogram.api.types), 285  
ChannelAdminLogEventActionChangeUsername (class in pyrogram.api.types), 285  
ChannelAdminLogEventActionDefaultBannedRights (class in pyrogram.api.types), 285  
ChannelAdminLogEventActionDeleteMessage (class in pyrogram.api.types), 286  
ChannelAdminLogEventActionEditMessage (class in pyrogram.api.types), 286  
ChannelAdminLogEventActionParticipantInvite (class in pyrogram.api.types), 286  
ChannelAdminLogEventActionParticipantJoin (class in pyrogram.api.types), 286  
ChannelAdminLogEventActionParticipantLeave (class in pyrogram.api.types), 287  
ChannelAdminLogEventActionParticipantToggleAdmin (class in pyrogram.api.types), 287  
ChannelAdminLogEventActionParticipantToggleBan (class in pyrogram.api.types), 287  
ChannelAdminLogEventActionStopPoll (class in pyrogram.api.types), 287  
ChannelAdminLogEventActionToggleInvites (class in pyrogram.api.types), 288  
ChannelAdminLogEventActionTogglePreHistoryHidden (class in pyrogram.api.types), 288  
ChannelAdminLogEventActionToggleSignatures (class in pyrogram.api.types), 288  
ChannelAdminLogEventActionUpdatePinned (class in pyrogram.api.types), 288  
ChannelAdminLogEventsFilter (class in pyrogram.api.types), 288  
ChannelDifference (class in pyrogram.api.types.updates), 270  
ChannelDifferenceEmpty (class in pyrogram.api.types.updates), 271  
ChannelDifferenceTooLong (class in pyrogram.api.types.updates), 271  
ChannelForbidden (class in pyrogram.api.types), 289  
ChannelFull (class in pyrogram.api.types), 289  
ChannelMessages (class in pyrogram.api.types.messages), 253  
ChannelMessagesFilter (class in pyrogram.api.types), 290  
ChannelMessagesFilterEmpty (class in pyrogram.api.types), 290  
ChannelParticipant (class in pyrogram.api.types), 291  
ChannelParticipant (class in pyrogram.api.types.channels), 240  
ChannelParticipantAdmin (class in pyrogram.api.types), 291  
ChannelParticipantBanned (class in pyrogram.api.types), 291  
ChannelParticipantCreator (class in pyrogram.api.types), 292  
ChannelParticipants (class in pyrogram.api.types.channels), 241  
ChannelParticipantsAdmins (class in pyrogram.api.types), 292  
ChannelParticipantsBanned (class in pyrogram

- gram.api.types*), 292
- ChannelParticipantsBots (class in *pyrogram.api.types*), 292
- ChannelParticipantsContacts (class in *pyrogram.api.types*), 293
- ChannelParticipantSelf (class in *pyrogram.api.types*), 292
- ChannelParticipantsKicked (class in *pyrogram.api.types*), 293
- ChannelParticipantsNotModified (class in *pyrogram.api.types.channels*), 241
- ChannelParticipantsRecent (class in *pyrogram.api.types*), 293
- ChannelParticipantsSearch (class in *pyrogram.api.types*), 293
- Chat (class in *pyrogram*), 58
- Chat (class in *pyrogram.api.types*), 293
- ChatAdminRights (class in *pyrogram.api.types*), 294
- ChatBannedRights (class in *pyrogram.api.types*), 294
- ChatEmpty (class in *pyrogram.api.types*), 295
- ChatForbidden (class in *pyrogram.api.types*), 295
- ChatFull (class in *pyrogram.api.types*), 295
- ChatFull (class in *pyrogram.api.types.messages*), 254
- ChatInvite (class in *pyrogram.api.types*), 296
- ChatInviteAlready (class in *pyrogram.api.types*), 296
- ChatInviteEmpty (class in *pyrogram.api.types*), 297
- ChatInviteExported (class in *pyrogram.api.types*), 297
- ChatMember (class in *pyrogram*), 59
- ChatMembers (class in *pyrogram*), 60
- ChatOnlines (class in *pyrogram.api.types*), 297
- ChatParticipant (class in *pyrogram.api.types*), 297
- ChatParticipantAdmin (class in *pyrogram.api.types*), 298
- ChatParticipantCreator (class in *pyrogram.api.types*), 298
- ChatParticipants (class in *pyrogram.api.types*), 298
- ChatParticipantsForbidden (class in *pyrogram.api.types*), 298
- ChatPermissions (class in *pyrogram*), 60
- ChatPhoto (class in *pyrogram*), 59
- ChatPhoto (class in *pyrogram.api.types*), 299
- ChatPhotoEmpty (class in *pyrogram.api.types*), 299
- ChatPreview (class in *pyrogram*), 59
- Chats (class in *pyrogram.api.types.messages*), 254
- ChatsSlice (class in *pyrogram.api.types.messages*), 254
- CheckChatInvite (class in *pyrogram.api.functions.messages*), 182
- CheckedPhone (class in *pyrogram.api.types.auth*), 237
- CheckPassword (class in *pyrogram.api.functions.auth*), 158
- CheckUsername (class in *pyrogram.api.functions.account*), 144
- CheckUsername (class in *pyrogram.api.functions.channels*), 162
- ClearAllDrafts (class in *pyrogram.api.functions.messages*), 183
- ClearRecentStickers (class in *pyrogram.api.functions.messages*), 183
- ClearSavedInfo (class in *pyrogram.api.functions.payments*), 217
- click() (*pyrogram.Message* method), 75
- Client (class in *pyrogram*), 18
- ClientDHInnerData (class in *pyrogram.api.types*), 299
- CodeSettings (class in *pyrogram.api.types*), 299
- CodeTypeCall (class in *pyrogram.api.types.auth*), 238
- CodeTypeFlashCall (class in *pyrogram.api.types.auth*), 238
- CodeTypeSms (class in *pyrogram.api.types.auth*), 238
- command() (*pyrogram.Filters* static method), 106
- Config (class in *pyrogram.api.types*), 300
- ConfigSimple (class in *pyrogram.api.types.help*), 246
- ConfirmCall (class in *pyrogram.api.functions.phone*), 219
- ConfirmPasswordEmail (class in *pyrogram.api.functions.account*), 144
- ConfirmPhone (class in *pyrogram.api.functions.account*), 145
- Contact (class in *pyrogram*), 67
- Contact (class in *pyrogram.api.types*), 301
- contact (*pyrogram.Filters* attribute), 104
- ContactBlocked (class in *pyrogram.api.types*), 301
- ContactLinkContact (class in *pyrogram.api.types*), 302
- ContactLinkHasPhone (class in *pyrogram.api.types*), 302
- ContactLinkNone (class in *pyrogram.api.types*), 302
- ContactLinkUnknown (class in *pyrogram.api.types*), 302
- Contacts (class in *pyrogram.api.types.contacts*), 242
- ContactsNotModified (class in *pyrogram.api.types.contacts*), 242
- ContactStatus (class in *pyrogram.api.types*), 302
- create() (*pyrogram.Filters* method), 103
- CreateChannel (class in *pyrogram.api.functions.channels*), 163
- CreateChat (class in *pyrogram.api.functions.messages*), 183
- CreateStickerSet (class in *pyrogram.api.functions.stickers*), 223

## D

- DataJSON (class in pyrogram.api.types), 303
- DC, 141
- DcOption (class in pyrogram.api.types), 303
- Decorator, 141
- DeepLinkInfo (class in pyrogram.api.types.help), 246
- DeepLinkInfoEmpty (class in pyrogram.api.types.help), 246
- delete() (pyrogram.Message method), 76
- delete\_chat\_photo (pyrogram.Filters attribute), 105
- delete\_chat\_photo() (pyrogram.Client method), 44
- delete\_contacts() (pyrogram.Client method), 51
- delete\_messages() (pyrogram.Client method), 37
- delete\_user\_profile\_photos() (pyrogram.Client method), 50
- DeleteAccount (class in pyrogram.api.functions.account), 145
- DeleteByPhones (class in pyrogram.api.functions.contacts), 171
- DeleteChannel (class in pyrogram.api.functions.channels), 163
- DeleteChatUser (class in pyrogram.api.functions.messages), 183
- DeleteContact (class in pyrogram.api.functions.contacts), 172
- DeleteContacts (class in pyrogram.api.functions.contacts), 172
- DeletedMessagesHandler (class in pyrogram), 99
- DeleteHistory (class in pyrogram.api.functions.channels), 163
- DeleteHistory (class in pyrogram.api.functions.messages), 184
- DeleteMessages (class in pyrogram.api.functions.channels), 163
- DeleteMessages (class in pyrogram.api.functions.messages), 184
- DeletePhotos (class in pyrogram.api.functions.photos), 221
- DeleteSecureValue (class in pyrogram.api.functions.account), 145
- DeleteUserHistory (class in pyrogram.api.functions.channels), 164
- DestroyAuthKey (class in pyrogram.api.functions), 228
- DestroyAuthKeyFail (class in pyrogram.api.types), 304
- DestroyAuthKeyNone (class in pyrogram.api.types), 304
- DestroyAuthKeyOk (class in pyrogram.api.types), 304
- DestroySession (class in pyrogram.api.functions), 228
- DestroySessionNone (class in pyrogram.api.types), 304
- DestroySessionOk (class in pyrogram.api.types), 305
- DhConfig (class in pyrogram.api.types.messages), 255
- DhConfigNotModified (class in pyrogram.api.types.messages), 255
- DhGenFail (class in pyrogram.api.types), 305
- DhGenOk (class in pyrogram.api.types), 305
- DhGenRetry (class in pyrogram.api.types), 306
- Dialog (class in pyrogram), 61
- Dialog (class in pyrogram.api.types), 306
- DialogPeer (class in pyrogram.api.types), 306
- Dialogs (class in pyrogram), 61
- Dialogs (class in pyrogram.api.types.messages), 256
- DialogsNotModified (class in pyrogram.api.types.messages), 256
- DialogsSlice (class in pyrogram.api.types.messages), 256
- Difference (class in pyrogram.api.types.updates), 272
- DifferenceEmpty (class in pyrogram.api.types.updates), 273
- DifferenceSlice (class in pyrogram.api.types.updates), 273
- DifferenceTooLong (class in pyrogram.api.types.updates), 274
- DiscardCall (class in pyrogram.api.functions.phone), 219
- DiscardEncryption (class in pyrogram.api.functions.messages), 184
- DisconnectHandler (class in pyrogram), 100
- Document (class in pyrogram), 65
- Document (class in pyrogram.api.types), 307
- document (pyrogram.Filters attribute), 104
- DocumentAttributeAnimated (class in pyrogram.api.types), 307
- DocumentAttributeAudio (class in pyrogram.api.types), 307
- DocumentAttributeFilename (class in pyrogram.api.types), 308
- DocumentAttributeHasStickers (class in pyrogram.api.types), 308
- DocumentAttributeImageSize (class in pyrogram.api.types), 308
- DocumentAttributeSticker (class in pyrogram.api.types), 308
- DocumentAttributeVideo (class in pyrogram.api.types), 309
- DocumentEmpty (class in pyrogram.api.types), 309
- download() (pyrogram.Message method), 76
- download\_media() (pyrogram.Client method), 41
- DraftMessage (class in pyrogram.api.types), 309
- DraftMessageEmpty (class in pyrogram.api.types), 309



310  
DropTempAuthKeys (class in pyrogram.api.functions.auth), 159

## E

edit() (pyrogram.Message method), 77  
edit\_caption() (pyrogram.Message method), 78  
edit\_media() (pyrogram.Message method), 78  
edit\_message\_caption() (pyrogram.Client method), 36  
edit\_message\_media() (pyrogram.Client method), 37  
edit\_message\_reply\_markup() (pyrogram.Client method), 37  
edit\_message\_text() (pyrogram.Client method), 36  
edit\_reply\_markup() (pyrogram.Message method), 79  
EditAdmin (class in pyrogram.api.functions.channels), 164  
EditBanned (class in pyrogram.api.functions.channels), 164  
EditChatAbout (class in pyrogram.api.functions.messages), 185  
EditChatAdmin (class in pyrogram.api.functions.messages), 185  
EditChatDefaultBannedRights (class in pyrogram.api.functions.messages), 185  
EditChatPhoto (class in pyrogram.api.functions.messages), 186  
EditChatTitle (class in pyrogram.api.functions.messages), 186  
edited (pyrogram.Filters attribute), 104  
EditInlineBotMessage (class in pyrogram.api.functions.messages), 186  
EditMessage (class in pyrogram.api.functions.messages), 187  
EditPhoto (class in pyrogram.api.functions.channels), 165  
EditTitle (class in pyrogram.api.functions.channels), 165  
EditUserInfo (class in pyrogram.api.functions.help), 176  
EmojiKeyword (class in pyrogram.api.types), 310  
EmojiKeywordDeleted (class in pyrogram.api.types), 310  
EmojiKeywordsDifference (class in pyrogram.api.types), 310  
EmojiURL (class in pyrogram.api.types), 311  
enable\_cloud\_password() (pyrogram.Client method), 51  
EncryptedChat (class in pyrogram.api.types), 311  
EncryptedChatDiscarded (class in pyrogram.api.types), 312

EncryptedChatEmpty (class in pyrogram.api.types), 312  
EncryptedChatRequested (class in pyrogram.api.types), 312  
EncryptedChatWaiting (class in pyrogram.api.types), 313  
EncryptedFile (class in pyrogram.api.types), 313  
EncryptedFileEmpty (class in pyrogram.api.types), 313  
EncryptedMessage (class in pyrogram.api.types), 314  
EncryptedMessageService (class in pyrogram.api.types), 314  
export\_chat\_invite\_link() (pyrogram.Client method), 44  
ExportAuthorization (class in pyrogram.api.functions.auth), 159  
ExportChatInvite (class in pyrogram.api.functions.messages), 188  
ExportedAuthorization (class in pyrogram.api.types.auth), 238  
ExportedMessageLink (class in pyrogram.api.types), 314  
ExportInvite (class in pyrogram.api.functions.channels), 165  
ExportMessageLink (class in pyrogram.api.functions.channels), 166

## F

FavedStickers (class in pyrogram.api.types.messages), 257  
FavedStickersNotModified (class in pyrogram.api.types.messages), 257  
FaveSticker (class in pyrogram.api.functions.messages), 188  
FeaturedStickers (class in pyrogram.api.types.messages), 257  
FeaturedStickersNotModified (class in pyrogram.api.types.messages), 258  
File (class in pyrogram.api.types.upload), 275  
FileCdnRedirect (class in pyrogram.api.types.upload), 276  
FileGif (class in pyrogram.api.types.storage), 268  
FileHash (class in pyrogram.api.types), 315  
FileJpeg (class in pyrogram.api.types.storage), 268  
FileLocation (class in pyrogram.api.types), 315  
FileLocationUnavailable (class in pyrogram.api.types), 315  
FileMov (class in pyrogram.api.types.storage), 268  
FileMp3 (class in pyrogram.api.types.storage), 268  
FileMp4 (class in pyrogram.api.types.storage), 268  
FilePartial (class in pyrogram.api.types.storage), 269  
FilePdf (class in pyrogram.api.types.storage), 269

FilePng (class in pyrogram.api.types.storage), 269  
 FileUnknown (class in pyrogram.api.types.storage), 269  
 FileWebp (class in pyrogram.api.types.storage), 269  
 Filters (class in pyrogram), 103  
 Filters.chat (class in pyrogram), 106  
 Filters.user (class in pyrogram), 106  
 FinishTakeoutSession (class in pyrogram.api.functions.account), 145  
 ForceReply (class in pyrogram), 70  
 forward() (pyrogram.Message method), 79  
 forward\_messages() (pyrogram.Client method), 25  
 forwarded (pyrogram.Filters attribute), 104  
 ForwardMessages (class in pyrogram.api.functions.messages), 188  
 Found (class in pyrogram.api.types.contacts), 243  
 FoundGif (class in pyrogram.api.types), 316  
 FoundGifCached (class in pyrogram.api.types), 316  
 FoundGifs (class in pyrogram.api.types.messages), 258  
 FoundStickerSets (class in pyrogram.api.types.messages), 258  
 FoundStickerSetsNotModified (class in pyrogram.api.types.messages), 258

## G

Game (class in pyrogram), 67  
 Game (class in pyrogram.api.types), 316  
 game (pyrogram.Filters attribute), 104  
 game\_high\_score (pyrogram.Filters attribute), 105  
 GameHighScore (class in pyrogram), 70  
 GameHighScores (class in pyrogram), 70  
 GeoPoint (class in pyrogram.api.types), 317  
 GeoPointEmpty (class in pyrogram.api.types), 317  
 get\_chat() (pyrogram.Client method), 46  
 get\_chat\_member() (pyrogram.Client method), 46  
 get\_chat\_members() (pyrogram.Client method), 46  
 get\_chat\_members\_count() (pyrogram.Client method), 47  
 get\_contacts() (pyrogram.Client method), 50  
 get\_contacts\_count() (pyrogram.Client method), 51  
 get\_dialogs() (pyrogram.Client method), 48  
 get\_dialogs\_count() (pyrogram.Client method), 48  
 get\_game\_high\_scores() (pyrogram.Client method), 55  
 get\_history() (pyrogram.Client method), 38  
 get\_history\_count() (pyrogram.Client method), 39  
 get\_inline\_bot\_results() (pyrogram.Client method), 52  
 get\_me() (pyrogram.Client method), 49  
 get\_messages() (pyrogram.Client method), 38

get\_user\_profile\_photos() (pyrogram.Client method), 49  
 get\_user\_profile\_photos\_count() (pyrogram.Client method), 50  
 get\_users() (pyrogram.Client method), 49  
 GetAccountTTL (class in pyrogram.api.functions.account), 146  
 GetAdmindedPublicChannels (class in pyrogram.api.functions.channels), 166  
 GetAdminLog (class in pyrogram.api.functions.channels), 166  
 GetAllChats (class in pyrogram.api.functions.messages), 189  
 GetAllDrafts (class in pyrogram.api.functions.messages), 189  
 GetAllSecureValues (class in pyrogram.api.functions.account), 146  
 GetAllStickers (class in pyrogram.api.functions.messages), 189  
 GetAppChangelog (class in pyrogram.api.functions.help), 176  
 GetAppConfig (class in pyrogram.api.functions.help), 177  
 GetAppUpdate (class in pyrogram.api.functions.help), 177  
 GetArchivedStickers (class in pyrogram.api.functions.messages), 189  
 GetAttachedStickers (class in pyrogram.api.functions.messages), 190  
 GetAuthorizationForm (class in pyrogram.api.functions.account), 146  
 GetAuthorizations (class in pyrogram.api.functions.account), 146  
 GetAutoDownloadSettings (class in pyrogram.api.functions.account), 147  
 GetBlocked (class in pyrogram.api.functions.contacts), 172  
 GetBotCallbackAnswer (class in pyrogram.api.functions.messages), 190  
 GetCallConfig (class in pyrogram.api.functions.phone), 220  
 GetCdnConfig (class in pyrogram.api.functions.help), 177  
 GetCdnFile (class in pyrogram.api.functions.upload), 224  
 GetCdnFileHashes (class in pyrogram.api.functions.upload), 225  
 GetChannelDifference (class in pyrogram.api.functions.updates), 223  
 GetChannels (class in pyrogram.api.functions.channels), 167  
 GetChats (class in pyrogram.api.functions.messages), 190  
 GetCommonChats (class in pyrogram



[gram.api.functions.messages](#)), 190  
[GetConfig](#) (class in [pyrogram.api.functions.help](#)), 177  
[GetContactIDs](#) (class in [pyrogram.api.functions.contacts](#)), 172  
[GetContacts](#) (class in [pyrogram.api.functions.contacts](#)), 173  
[GetContactSignUpNotification](#) (class in [pyrogram.api.functions.account](#)), 147  
[GetDeepLinkInfo](#) (class in [pyrogram.api.functions.help](#)), 178  
[GetDhConfig](#) (class in [pyrogram.api.functions.messages](#)), 191  
[GetDialogs](#) (class in [pyrogram.api.functions.messages](#)), 191  
[GetDialogUnreadMarks](#) (class in [pyrogram.api.functions.messages](#)), 191  
[GetDifference](#) (class in [pyrogram.api.functions.langpack](#)), 180  
[GetDifference](#) (class in [pyrogram.api.functions.updates](#)), 224  
[GetDocumentByHash](#) (class in [pyrogram.api.functions.messages](#)), 192  
[GetEmojiKeywords](#) (class in [pyrogram.api.functions.messages](#)), 192  
[GetEmojiKeywordsDifference](#) (class in [pyrogram.api.functions.messages](#)), 192  
[GetEmojiURL](#) (class in [pyrogram.api.functions.messages](#)), 192  
[GetFavedStickers](#) (class in [pyrogram.api.functions.messages](#)), 193  
[GetFeaturedStickers](#) (class in [pyrogram.api.functions.messages](#)), 193  
[GetFile](#) (class in [pyrogram.api.functions.upload](#)), 225  
[GetFileHashes](#) (class in [pyrogram.api.functions.upload](#)), 225  
[GetFullChannel](#) (class in [pyrogram.api.functions.channels](#)), 167  
[GetFullChat](#) (class in [pyrogram.api.functions.messages](#)), 193  
[GetFullUser](#) (class in [pyrogram.api.functions.users](#)), 227  
[GetFutureSalts](#) (class in [pyrogram.api.functions](#)), 228  
[GetGameHighScores](#) (class in [pyrogram.api.functions.messages](#)), 193  
[GetHistory](#) (class in [pyrogram.api.functions.messages](#)), 194  
[GetInlineBotResults](#) (class in [pyrogram.api.functions.messages](#)), 194  
[GetInlineGameHighScores](#) (class in [pyrogram.api.functions.messages](#)), 194  
[GetInviteText](#) (class in [pyrogram.api.functions.help](#)), 178  
[GetLangPack](#) (class in [pyrogram.api.functions.langpack](#)), 181  
[GetLanguage](#) (class in [pyrogram.api.functions.langpack](#)), 181  
[GetLanguages](#) (class in [pyrogram.api.functions.langpack](#)), 181  
[GetLeftChannels](#) (class in [pyrogram.api.functions.channels](#)), 167  
[GetMaskStickers](#) (class in [pyrogram.api.functions.messages](#)), 195  
[GetMessageEditData](#) (class in [pyrogram.api.functions.messages](#)), 195  
[GetMessages](#) (class in [pyrogram.api.functions.channels](#)), 167  
[GetMessages](#) (class in [pyrogram.api.functions.messages](#)), 195  
[GetMessagesViews](#) (class in [pyrogram.api.functions.messages](#)), 195  
[GetNearestDc](#) (class in [pyrogram.api.functions.help](#)), 178  
[GetNotifyExceptions](#) (class in [pyrogram.api.functions.account](#)), 147  
[GetNotifySettings](#) (class in [pyrogram.api.functions.account](#)), 147  
[GetOnlines](#) (class in [pyrogram.api.functions.messages](#)), 196  
[GetParticipant](#) (class in [pyrogram.api.functions.channels](#)), 168  
[GetParticipants](#) (class in [pyrogram.api.functions.channels](#)), 168  
[GetPassportConfig](#) (class in [pyrogram.api.functions.help](#)), 178  
[GetPassword](#) (class in [pyrogram.api.functions.account](#)), 148  
[GetPasswordSettings](#) (class in [pyrogram.api.functions.account](#)), 148  
[GetPaymentForm](#) (class in [pyrogram.api.functions.payments](#)), 217  
[GetPaymentReceipt](#) (class in [pyrogram.api.functions.payments](#)), 218  
[GetPeerDialogs](#) (class in [pyrogram.api.functions.messages](#)), 196  
[GetPeerSettings](#) (class in [pyrogram.api.functions.messages](#)), 196  
[GetPinnedDialogs](#) (class in [pyrogram.api.functions.messages](#)), 197  
[GetPollResults](#) (class in [pyrogram.api.functions.messages](#)), 197  
[GetPrivacy](#) (class in [pyrogram.api.functions.account](#)), 148  
[GetProxyData](#) (class in [pyrogram.api.functions.help](#)), 178  
[GetRecentLocations](#) (class in [pyrogram.api.functions.messages](#)), 197  
[GetRecentMeUrls](#) (class in [pyrogram](#)

- gram.api.functions.help*), 179
- GetRecentStickers (class in *pyrogram.api.functions.messages*), 197
- GetSaved (class in *pyrogram.api.functions.contacts*), 173
- GetSavedGifs (class in *pyrogram.api.functions.messages*), 198
- GetSavedInfo (class in *pyrogram.api.functions.payments*), 218
- GetSecureValue (class in *pyrogram.api.functions.account*), 148
- GetSplitRanges (class in *pyrogram.api.functions.messages*), 198
- GetState (class in *pyrogram.api.functions.updates*), 224
- GetStatsURL (class in *pyrogram.api.functions.messages*), 198
- GetStatuses (class in *pyrogram.api.functions.contacts*), 173
- GetStickers (class in *pyrogram.api.functions.messages*), 199
- GetStickerSet (class in *pyrogram.api.functions.messages*), 198
- GetStrings (class in *pyrogram.api.functions.langpack*), 181
- GetSupport (class in *pyrogram.api.functions.help*), 179
- GetSupportName (class in *pyrogram.api.functions.help*), 179
- GetTermsOfServiceUpdate (class in *pyrogram.api.functions.help*), 179
- GetTmpPassword (class in *pyrogram.api.functions.account*), 149
- GetTopPeers (class in *pyrogram.api.functions.contacts*), 173
- GetUnreadMentions (class in *pyrogram.api.functions.messages*), 199
- GetUserInfo (class in *pyrogram.api.functions.help*), 180
- GetUserPhotos (class in *pyrogram.api.functions.photos*), 221
- GetUsers (class in *pyrogram.api.functions.users*), 227
- GetWallPaper (class in *pyrogram.api.functions.account*), 149
- GetWallPapers (class in *pyrogram.api.functions.account*), 149
- GetWebAuthorizations (class in *pyrogram.api.functions.account*), 149
- GetWebFile (class in *pyrogram.api.functions.upload*), 226
- GetWebPage (class in *pyrogram.api.functions.messages*), 199
- GetWebPagePreview (class in *pyrogram.api.functions.messages*), 200
- group (*pyrogram.Filters* attribute), 105
- group\_chat\_created (*pyrogram.Filters* attribute), 105
- ## H
- Handler, 141
- HideReportSpam (class in *pyrogram.api.functions.messages*), 200
- HighScore (class in *pyrogram.api.types*), 317
- HighScores (class in *pyrogram.api.types.messages*), 259
- HttpWait (class in *pyrogram.api.types*), 317
- ## I
- ID (*pyrogram.api.functions.account.AcceptAuthorization* attribute), 143
- ID (*pyrogram.api.functions.account.CancelPasswordEmail* attribute), 144
- ID (*pyrogram.api.functions.account.ChangePhone* attribute), 144
- ID (*pyrogram.api.functions.account.CheckUsername* attribute), 144
- ID (*pyrogram.api.functions.account.ConfirmPasswordEmail* attribute), 144
- ID (*pyrogram.api.functions.account.ConfirmPhone* attribute), 145
- ID (*pyrogram.api.functions.account.DeleteAccount* attribute), 145
- ID (*pyrogram.api.functions.account.DeleteSecureValue* attribute), 145
- ID (*pyrogram.api.functions.account.FinishTakeoutSession* attribute), 145
- ID (*pyrogram.api.functions.account.GetAccountTTL* attribute), 146
- ID (*pyrogram.api.functions.account.GetAllSecureValues* attribute), 146
- ID (*pyrogram.api.functions.account.GetAuthorizationForm* attribute), 146
- ID (*pyrogram.api.functions.account.GetAuthorizations* attribute), 146
- ID (*pyrogram.api.functions.account.GetAutoDownloadSettings* attribute), 147
- ID (*pyrogram.api.functions.account.GetContactSignUpNotification* attribute), 147
- ID (*pyrogram.api.functions.account.GetNotifyExceptions* attribute), 147
- ID (*pyrogram.api.functions.account.GetNotifySettings* attribute), 147
- ID (*pyrogram.api.functions.account.GetPassword* attribute), 148
- ID (*pyrogram.api.functions.account.GetPasswordSettings* attribute), 148
- ID (*pyrogram.api.functions.account.GetPrivacy* attribute), 148

|                                                                                 |                                                                           |
|---------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| ID (pyrogram.api.functions.account.GetSecureValue attribute), 148               | ID (pyrogram.api.functions.account.UpdateNotifySettings attribute), 155   |
| ID (pyrogram.api.functions.account.GetTmpPassword attribute), 149               | ID (pyrogram.api.functions.account.UpdatePasswordSettings attribute), 156 |
| ID (pyrogram.api.functions.account.GetWallPaper attribute), 149                 | ID (pyrogram.api.functions.account.UpdateProfile attribute), 156          |
| ID (pyrogram.api.functions.account.GetWallPapers attribute), 149                | ID (pyrogram.api.functions.account.UpdateStatus attribute), 156           |
| ID (pyrogram.api.functions.account.GetWebAuthorizations attribute), 149         | ID (pyrogram.api.functions.account.UpdateUsername attribute), 157         |
| ID (pyrogram.api.functions.account.InitTakeoutSession attribute), 150           | ID (pyrogram.api.functions.account.UploadWallPaper attribute), 157        |
| ID (pyrogram.api.functions.account.InstallWallPaper attribute), 150             | ID (pyrogram.api.functions.account.VerifyEmail attribute), 157            |
| ID (pyrogram.api.functions.account.RegisterDevice attribute), 150               | ID (pyrogram.api.functions.account.VerifyPhone attribute), 157            |
| ID (pyrogram.api.functions.account.ReportPeer attribute), 151                   | ID (pyrogram.api.functions.auth.BindTempAuthKey attribute), 158           |
| ID (pyrogram.api.functions.account.ResendPasswordEmail attribute), 151          | ID (pyrogram.api.functions.auth.CancelCode attribute), 158                |
| ID (pyrogram.api.functions.account.ResetAuthorization attribute), 151           | ID (pyrogram.api.functions.auth.CheckPassword attribute), 158             |
| ID (pyrogram.api.functions.account.ResetNotifySettings attribute), 151          | ID (pyrogram.api.functions.auth.DropTempAuthKeys attribute), 159          |
| ID (pyrogram.api.functions.account.ResetWallPapers attribute), 152              | ID (pyrogram.api.functions.auth.ExportAuthorization attribute), 159       |
| ID (pyrogram.api.functions.account.ResetWebAuthorization attribute), 152        | ID (pyrogram.api.functions.auth.ImportAuthorization attribute), 159       |
| ID (pyrogram.api.functions.account.ResetWebAuthorizations attribute), 152       | ID (pyrogram.api.functions.auth.ImportBotAuthorization attribute), 159    |
| ID (pyrogram.api.functions.account.SaveAutoDownloadSettings attribute), 152     | ID (pyrogram.api.functions.auth.LogOut attribute), 160                    |
| ID (pyrogram.api.functions.account.SaveSecureValue attribute), 153              | ID (pyrogram.api.functions.auth.RecoverPassword attribute), 160           |
| ID (pyrogram.api.functions.account.SaveWallPaper attribute), 153                | ID (pyrogram.api.functions.auth.RequestPasswordRecovery attribute), 160   |
| ID (pyrogram.api.functions.account.SendChangePhoneCode attribute), 153          | ID (pyrogram.api.functions.auth.ResendCode attribute), 160                |
| ID (pyrogram.api.functions.account.SendConfirmPhoneCode attribute), 153         | ID (pyrogram.api.functions.auth.ResetAuthorizations attribute), 160       |
| ID (pyrogram.api.functions.account.SendVerifyEmailCode attribute), 154          | ID (pyrogram.api.functions.auth.SendCode attribute), 161                  |
| ID (pyrogram.api.functions.account.SendVerifyPhoneCode attribute), 154          | ID (pyrogram.api.functions.auth.SignIn attribute), 161                    |
| ID (pyrogram.api.functions.account.SetAccountTTL attribute), 154                | ID (pyrogram.api.functions.auth.SignUp attribute), 161                    |
| ID (pyrogram.api.functions.account.SetContactSignUpNotification attribute), 154 | ID (pyrogram.api.functions.bots.AnswerWebhookJSONQuery attribute), 162    |
| ID (pyrogram.api.functions.account.SetPrivacy attribute), 155                   | ID (pyrogram.api.functions.bots.SendCustomRequest attribute), 162         |
| ID (pyrogram.api.functions.account.UnregisterDevice attribute), 155             | ID (pyrogram.api.functions.channels.CheckUsername attribute), 162         |
| ID (pyrogram.api.functions.account.UpdateDeviceLocked attribute), 155           | ID (pyrogram.api.functions.channels.CreateChannel attribute), 163         |
|                                                                                 | ID (pyrogram.api.functions.channels.DeleteChannel attribute), 163         |
|                                                                                 | ID (pyrogram.api.functions.channels.DeleteHistory attribute), 163         |

|                                                                            |                                                                        |
|----------------------------------------------------------------------------|------------------------------------------------------------------------|
| tribute), 163                                                              | 171                                                                    |
| ID (pyrogram.api.functions.channels.DeleteMessages attribute), 163         | ID (pyrogram.api.functions.contacts.DeleteByPhones attribute), 171     |
| ID (pyrogram.api.functions.channels.DeleteUserHistory attribute), 164      | ID (pyrogram.api.functions.contacts.DeleteContact attribute), 172      |
| ID (pyrogram.api.functions.channels.EditAdmin attribute), 164              | ID (pyrogram.api.functions.contacts.DeleteContacts attribute), 172     |
| ID (pyrogram.api.functions.channels.EditBanned attribute), 164             | ID (pyrogram.api.functions.contacts.GetBlocked attribute), 172         |
| ID (pyrogram.api.functions.channels.EditPhoto attribute), 165              | ID (pyrogram.api.functions.contacts.GetContactIDs attribute), 172      |
| ID (pyrogram.api.functions.channels.EditTitle attribute), 165              | ID (pyrogram.api.functions.contacts.GetContacts attribute), 173        |
| ID (pyrogram.api.functions.channels.ExportInvite attribute), 165           | ID (pyrogram.api.functions.contacts.GetSaved attribute), 173           |
| ID (pyrogram.api.functions.channels.ExportMessageLink attribute), 166      | ID (pyrogram.api.functions.contacts.GetStatuses attribute), 173        |
| ID (pyrogram.api.functions.channels.GetAdminPublicChannel attribute), 166  | ID (pyrogram.api.functions.contacts.GetTopPeers attribute), 173        |
| ID (pyrogram.api.functions.channels.GetAdminLog attribute), 166            | ID (pyrogram.api.functions.contacts.ImportContacts attribute), 174     |
| ID (pyrogram.api.functions.channels.GetChannels attribute), 167            | ID (pyrogram.api.functions.contacts.ResetSaved attribute), 174         |
| ID (pyrogram.api.functions.channels.GetFullChannel attribute), 167         | ID (pyrogram.api.functions.contacts.ResetTopPeerRating attribute), 174 |
| ID (pyrogram.api.functions.channels.GetLeftChannels attribute), 167        | ID (pyrogram.api.functions.contacts.ResolveUsername attribute), 174    |
| ID (pyrogram.api.functions.channels.GetMessages attribute), 167            | ID (pyrogram.api.functions.contacts.Search attribute), 175             |
| ID (pyrogram.api.functions.channels.GetParticipant attribute), 168         | ID (pyrogram.api.functions.contacts.ToggleTopPeers attribute), 175     |
| ID (pyrogram.api.functions.channels.GetParticipants attribute), 168        | ID (pyrogram.api.functions.contacts.Unblock attribute), 175            |
| ID (pyrogram.api.functions.channels.InviteToChannel attribute), 168        | ID (pyrogram.api.functions.contest.SaveDeveloperInfo attribute), 175   |
| ID (pyrogram.api.functions.channels.JoinChannel attribute), 169            | ID (pyrogram.api.functions.DestroyAuthKey attribute), 228              |
| ID (pyrogram.api.functions.channels.LeaveChannel attribute), 169           | ID (pyrogram.api.functions.DestroySession attribute), 228              |
| ID (pyrogram.api.functions.channels.ReadHistory attribute), 169            | ID (pyrogram.api.functions.GetFutureSalts attribute), 228              |
| ID (pyrogram.api.functions.channels.ReadMessageContents attribute), 169    | ID (pyrogram.api.functions.help.AcceptTermsOfService attribute), 176   |
| ID (pyrogram.api.functions.channels.ReportSpam attribute), 170             | ID (pyrogram.api.functions.help.EditUserInfo attribute), 176           |
| ID (pyrogram.api.functions.channels.SetStickers attribute), 170            | ID (pyrogram.api.functions.help.GetAppChangelog attribute), 176        |
| ID (pyrogram.api.functions.channels.TogglePreHistoryHidden attribute), 170 | ID (pyrogram.api.functions.help.GetAppConfig attribute), 177           |
| ID (pyrogram.api.functions.channels.ToggleSignatures attribute), 171       | ID (pyrogram.api.functions.help.GetAppUpdate attribute), 177           |
| ID (pyrogram.api.functions.channels.UpdateUsername attribute), 171         | ID (pyrogram.api.functions.help.GetCdnConfig attribute), 177           |
| ID (pyrogram.api.functions.contacts.Block attribute), 171                  | ID (pyrogram.api.functions.help.GetConfig attribute), 177              |

|    |                                                                      |    |                                                                              |
|----|----------------------------------------------------------------------|----|------------------------------------------------------------------------------|
|    | 177                                                                  |    | attribute), 182                                                              |
| ID | (pyrogram.api.functions.help.GetDeepLinkInfo attribute), 178         | ID | (pyrogram.api.functions.messages.ClearAllDrafts attribute), 183              |
| ID | (pyrogram.api.functions.help.GetInviteText attribute), 178           | ID | (pyrogram.api.functions.messages.ClearRecentStickers attribute), 183         |
| ID | (pyrogram.api.functions.help.GetNearestDc attribute), 178            | ID | (pyrogram.api.functions.messages.CreateChat attribute), 183                  |
| ID | (pyrogram.api.functions.help.GetPassportConfig attribute), 178       | ID | (pyrogram.api.functions.messages.DeleteChatUser attribute), 183              |
| ID | (pyrogram.api.functions.help.GetProxyData attribute), 178            | ID | (pyrogram.api.functions.messages.DeleteHistory attribute), 184               |
| ID | (pyrogram.api.functions.help.GetRecentMeUrls attribute), 179         | ID | (pyrogram.api.functions.messages.DeleteMessages attribute), 184              |
| ID | (pyrogram.api.functions.help.GetSupport attribute), 179              | ID | (pyrogram.api.functions.messages.DiscardEncryption attribute), 184           |
| ID | (pyrogram.api.functions.help.GetSupportName attribute), 179          | ID | (pyrogram.api.functions.messages.EditChatAbout attribute), 185               |
| ID | (pyrogram.api.functions.help.GetTermsOfServiceUpdate attribute), 179 | ID | (pyrogram.api.functions.messages.EditChatAdmin attribute), 185               |
| ID | (pyrogram.api.functions.help.GetUserInfo attribute), 180             | ID | (pyrogram.api.functions.messages.EditChatDefaultBannedRights attribute), 185 |
| ID | (pyrogram.api.functions.help.SaveAppLog attribute), 180              | ID | (pyrogram.api.functions.messages.EditChatPhoto attribute), 186               |
| ID | (pyrogram.api.functions.help.SetBotUpdatesStatus attribute), 180     | ID | (pyrogram.api.functions.messages.EditChatTitle attribute), 186               |
| ID | (pyrogram.api.functions.InitConnection attribute), 228               | ID | (pyrogram.api.functions.messages.EditInlineBotMessage attribute), 186        |
| ID | (pyrogram.api.functions.InvokeAfterMsg attribute), 229               | ID | (pyrogram.api.functions.messages.EditMessage attribute), 187                 |
| ID | (pyrogram.api.functions.InvokeAfterMsgs attribute), 229              | ID | (pyrogram.api.functions.messages.ExportChatInvite attribute), 188            |
| ID | (pyrogram.api.functions.InvokeWithLayer attribute), 229              | ID | (pyrogram.api.functions.messages.FaveSticker attribute), 188                 |
| ID | (pyrogram.api.functions.InvokeWithMessagesRange attribute), 230      | ID | (pyrogram.api.functions.messages.ForwardMessages attribute), 188             |
| ID | (pyrogram.api.functions.InvokeWithoutUpdates attribute), 230         | ID | (pyrogram.api.functions.messages.GetAllChats attribute), 189                 |
| ID | (pyrogram.api.functions.InvokeWithTakeout attribute), 230            | ID | (pyrogram.api.functions.messages.GetAllDrafts attribute), 189                |
| ID | (pyrogram.api.functions.langpack.GetDifference attribute), 180       | ID | (pyrogram.api.functions.messages.GetAllStickers attribute), 189              |
| ID | (pyrogram.api.functions.langpack.GetLangPack attribute), 181         | ID | (pyrogram.api.functions.messages.GetArchivedStickers attribute), 189         |
| ID | (pyrogram.api.functions.langpack.GetLanguage attribute), 181         | ID | (pyrogram.api.functions.messages.GetAttachedStickers attribute), 190         |
| ID | (pyrogram.api.functions.langpack.GetLanguages attribute), 181        | ID | (pyrogram.api.functions.messages.GetBotCallbackAnswer attribute), 190        |
| ID | (pyrogram.api.functions.langpack.GetStrings attribute), 181          | ID | (pyrogram.api.functions.messages.GetChats attribute), 190                    |
| ID | (pyrogram.api.functions.messages.AcceptEncryption attribute), 182    | ID | (pyrogram.api.functions.messages.GetCommonChats attribute), 190              |
| ID | (pyrogram.api.functions.messages.AddChatUser attribute), 182         | ID | (pyrogram.api.functions.messages.GetDhConfig attribute), 191                 |
| ID | (pyrogram.api.functions.messages.CheckChatInvite                     | ID | (pyrogram.api.functions.messages.GetDialogs at-                              |



|                                                                                |                                                                          |
|--------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| tribute), 191                                                                  | tribute), 199                                                            |
| ID (pyrogram.api.functions.messages.GetDialogUnreadMark attribute), 191        | ID (pyrogram.api.functions.messages.GetStickerSet attribute), 198        |
| ID (pyrogram.api.functions.messages.GetDocumentByHash attribute), 192          | ID (pyrogram.api.functions.messages.GetUnreadMentions attribute), 199    |
| ID (pyrogram.api.functions.messages.GetEmojiKeywords attribute), 192           | ID (pyrogram.api.functions.messages.GetWebPage attribute), 199           |
| ID (pyrogram.api.functions.messages.GetEmojiKeywordsDifference attribute), 192 | ID (pyrogram.api.functions.messages.GetWebPagePreview attribute), 200    |
| ID (pyrogram.api.functions.messages.GetEmojiURL attribute), 192                | ID (pyrogram.api.functions.messages.HideReportSpam attribute), 200       |
| ID (pyrogram.api.functions.messages.GetFavedStickers attribute), 193           | ID (pyrogram.api.functions.messages.ImportChatInvite attribute), 200     |
| ID (pyrogram.api.functions.messages.GetFeaturedStickers attribute), 193        | ID (pyrogram.api.functions.messages.InstallStickerSet attribute), 201    |
| ID (pyrogram.api.functions.messages.GetFullChat attribute), 193                | ID (pyrogram.api.functions.messages.MarkDialogUnread attribute), 201     |
| ID (pyrogram.api.functions.messages.GetGameHighScores attribute), 193          | ID (pyrogram.api.functions.messages.MigrateChat attribute), 201          |
| ID (pyrogram.api.functions.messages.GetHistory attribute), 194                 | ID (pyrogram.api.functions.messages.ReadEncryptedHistory attribute), 202 |
| ID (pyrogram.api.functions.messages.GetInlineBotResults attribute), 194        | ID (pyrogram.api.functions.messages.ReadFeaturedStickers attribute), 202 |
| ID (pyrogram.api.functions.messages.GetInlineGameHighScores attribute), 194    | ID (pyrogram.api.functions.messages.ReadHistory attribute), 202          |
| ID (pyrogram.api.functions.messages.GetMaskStickers attribute), 195            | ID (pyrogram.api.functions.messages.ReadMentions attribute), 202         |
| ID (pyrogram.api.functions.messages.GetMessageEditData attribute), 195         | ID (pyrogram.api.functions.messages.ReadMessageContents attribute), 203  |
| ID (pyrogram.api.functions.messages.GetMessages attribute), 195                | ID (pyrogram.api.functions.messages.ReceivedMessages attribute), 203     |
| ID (pyrogram.api.functions.messages.GetMessagesViews attribute), 195           | ID (pyrogram.api.functions.messages.ReceivedQueue attribute), 203        |
| ID (pyrogram.api.functions.messages.GetOnlines attribute), 196                 | ID (pyrogram.api.functions.messages.ReorderPinnedDialogs attribute), 203 |
| ID (pyrogram.api.functions.messages.GetPeerDialogs attribute), 196             | ID (pyrogram.api.functions.messages.ReorderStickerSets attribute), 204   |
| ID (pyrogram.api.functions.messages.GetPeerSettings attribute), 196            | ID (pyrogram.api.functions.messages.Report attribute), 204               |
| ID (pyrogram.api.functions.messages.GetPinnedDialogs attribute), 197           | ID (pyrogram.api.functions.messages.ReportEncryptedSpam attribute), 204  |
| ID (pyrogram.api.functions.messages.GetPollResults attribute), 197             | ID (pyrogram.api.functions.messages.ReportSpam attribute), 204           |
| ID (pyrogram.api.functions.messages.GetRecentLocations attribute), 197         | ID (pyrogram.api.functions.messages.RequestEncryption attribute), 205    |
| ID (pyrogram.api.functions.messages.GetRecentStickers attribute), 197          | ID (pyrogram.api.functions.messages.SaveDraft attribute), 205            |
| ID (pyrogram.api.functions.messages.GetSavedGifs attribute), 198               | ID (pyrogram.api.functions.messages.SaveGif attribute), 205              |
| ID (pyrogram.api.functions.messages.GetSplitRanges attribute), 198             | ID (pyrogram.api.functions.messages.SaveRecentSticker attribute), 206    |
| ID (pyrogram.api.functions.messages.GetStatsURL attribute), 198                | ID (pyrogram.api.functions.messages.Search attribute), 206               |
| ID (pyrogram.api.functions.messages.GetStickers attribute), 198                | ID (pyrogram.api.functions.messages.SearchGifs attribute), 206           |

|                                                                                   |                                                                           |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| tribute), 207                                                                     | attribute), 217                                                           |
| ID (pyrogram.api.functions.messages.SearchGlobal attribute), 207                  | ID (pyrogram.api.functions.payments.GetPaymentReceipt attribute), 218     |
| ID (pyrogram.api.functions.messages.SearchStickerSets attribute), 207             | ID (pyrogram.api.functions.payments.GetSavedInfo attribute), 218          |
| ID (pyrogram.api.functions.messages.SendEncrypted attribute), 208                 | ID (pyrogram.api.functions.payments.SendPaymentForm attribute), 218       |
| ID (pyrogram.api.functions.messages.SendEncryptedFile attribute), 208             | ID (pyrogram.api.functions.payments.ValidateRequestedInfo attribute), 218 |
| ID (pyrogram.api.functions.messages.SendEncryptedServiceAttribute attribute), 209 | ID (pyrogram.api.functions.phone.AcceptCall attribute), 219               |
| ID (pyrogram.api.functions.messages.SendInlineBotResult attribute), 209           | ID (pyrogram.api.functions.phone.ConfirmCall attribute), 219              |
| ID (pyrogram.api.functions.messages.SendMedia attribute), 209                     | ID (pyrogram.api.functions.phone.DiscardCall attribute), 219              |
| ID (pyrogram.api.functions.messages.SendMessage attribute), 210                   | ID (pyrogram.api.functions.phone.GetCallConfig attribute), 220            |
| ID (pyrogram.api.functions.messages.SendMultiMedia attribute), 211                | ID (pyrogram.api.functions.phone.ReceivedCall attribute), 220             |
| ID (pyrogram.api.functions.messages.SendScreenshotNotification attribute), 211    | ID (pyrogram.api.functions.phone.RequestCall attribute), 220              |
| ID (pyrogram.api.functions.messages.SendVote attribute), 212                      | ID (pyrogram.api.functions.phone.SaveCallDebug attribute), 221            |
| ID (pyrogram.api.functions.messages.SetBotCallbackAnswer attribute), 212          | ID (pyrogram.api.functions.phone.SetCallRating attribute), 221            |
| ID (pyrogram.api.functions.messages.SetBotPrecheckoutResults attribute), 212      | ID (pyrogram.api.functions.photos.DeletePhotos attribute), 221            |
| ID (pyrogram.api.functions.messages.SetBotShippingResults attribute), 213         | ID (pyrogram.api.functions.photos.GetUserPhotos attribute), 221           |
| ID (pyrogram.api.functions.messages.SetEncryptedTyping attribute), 213            | ID (pyrogram.api.functions.photos.UpdateProfilePhoto attribute), 222      |
| ID (pyrogram.api.functions.messages.SetGameScore attribute), 213                  | ID (pyrogram.api.functions.photos.UploadProfilePhoto attribute), 222      |
| ID (pyrogram.api.functions.messages.SetInlineBotResults attribute), 214           | ID (pyrogram.api.functions.Ping attribute), 231                           |
| ID (pyrogram.api.functions.messages.SetInlineGameScore attribute), 214            | ID (pyrogram.api.functions.PingDelayDisconnect attribute), 231            |
| ID (pyrogram.api.functions.messages.SetTyping attribute), 215                     | ID (pyrogram.api.functions.ReqDHParams attribute), 231                    |
| ID (pyrogram.api.functions.messages.StartBot attribute), 215                      | ID (pyrogram.api.functions.ReqPq attribute), 231                          |
| ID (pyrogram.api.functions.messages.ToggleDialogPin attribute), 215               | ID (pyrogram.api.functions.ReqPqMulti attribute), 232                     |
| ID (pyrogram.api.functions.messages.UninstallStickerSet attribute), 216           | ID (pyrogram.api.functions.RpcDropAnswer attribute), 232                  |
| ID (pyrogram.api.functions.messages.UpdatePinnedMessage attribute), 216           | ID (pyrogram.api.functions.SetClientDHParams attribute), 232              |
| ID (pyrogram.api.functions.messages.UploadEncryptedFile attribute), 216           | ID (pyrogram.api.functions.stickers.AddStickerToSet attribute), 222       |
| ID (pyrogram.api.functions.messages.UploadMedia attribute), 217                   | ID (pyrogram.api.functions.stickers.ChangeStickerPosition attribute), 223 |
| ID (pyrogram.api.functions.payments.ClearSavedInfo attribute), 217                | ID (pyrogram.api.functions.stickers.CreateStickerSet attribute), 223      |
| ID (pyrogram.api.functions.payments.GetPaymentForm attribute), 217                | ID (pyrogram.api.functions.stickers.RemoveStickerFromSet attribute), 223  |
|                                                                                   | ID (pyrogram.api.functions.updates.GetChannelDifference attribute), 223   |

|                                                                       |                                                                                   |
|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| ID (pyrogram.api.functions.updates.GetDifference attribute), 224      | ID (pyrogram.api.types.auth.CheckedPhone attribute), 237                          |
| ID (pyrogram.api.functions.updates.GetState attribute), 224           | ID (pyrogram.api.types.auth.CodeTypeCall attribute), 238                          |
| ID (pyrogram.api.functions.upload.GetCdnFile attribute), 224          | ID (pyrogram.api.types.auth.CodeTypeFlashCall attribute), 238                     |
| ID (pyrogram.api.functions.upload.GetCdnFileHashes attribute), 225    | ID (pyrogram.api.types.auth.CodeTypeSms attribute), 238                           |
| ID (pyrogram.api.functions.upload.GetFile attribute), 225             | ID (pyrogram.api.types.auth.ExportedAuthorization attribute), 238                 |
| ID (pyrogram.api.functions.upload.GetFileHashes attribute), 225       | ID (pyrogram.api.types.auth.PasswordRecovery attribute), 238                      |
| ID (pyrogram.api.functions.upload.GetWebFile attribute), 226          | ID (pyrogram.api.types.auth.SentCode attribute), 239                              |
| ID (pyrogram.api.functions.upload.ReuploadCdnFile attribute), 226     | ID (pyrogram.api.types.auth.SentCodeTypeApp attribute), 239                       |
| ID (pyrogram.api.functions.upload.SaveBigFilePart attribute), 226     | ID (pyrogram.api.types.auth.SentCodeTypeCall attribute), 239                      |
| ID (pyrogram.api.functions.upload.SaveFilePart attribute), 227        | ID (pyrogram.api.types.auth.SentCodeTypeFlashCall attribute), 240                 |
| ID (pyrogram.api.functions.users.GetFullUser attribute), 227          | ID (pyrogram.api.types.auth.SentCodeTypeSms attribute), 240                       |
| ID (pyrogram.api.functions.users.GetUsers attribute), 227             | ID (pyrogram.api.types.Authorization attribute), 277                              |
| ID (pyrogram.api.functions.users.SetSecureValueErrors attribute), 227 | ID (pyrogram.api.types.AutoDownloadSettings attribute), 278                       |
| ID (pyrogram.api.types.AccessPointRule attribute), 276                | ID (pyrogram.api.types.BadMsgNotification attribute), 278                         |
| ID (pyrogram.api.types.account.AuthorizationForm attribute), 233      | ID (pyrogram.api.types.BadServerSalt attribute), 278                              |
| ID (pyrogram.api.types.account.Authorizations attribute), 233         | ID (pyrogram.api.types.BindAuthKeyInner attribute), 279                           |
| ID (pyrogram.api.types.account.AutoDownloadSettings attribute), 233   | ID (pyrogram.api.types.BotCommand attribute), 279                                 |
| ID (pyrogram.api.types.account.Password attribute), 234               | ID (pyrogram.api.types.BotInfo attribute), 279                                    |
| ID (pyrogram.api.types.account.PasswordInputSettings attribute), 234  | ID (pyrogram.api.types.BotInlineMediaResult attribute), 279                       |
| ID (pyrogram.api.types.account.PasswordSettings attribute), 235       | ID (pyrogram.api.types.BotInlineMessageMediaAuto attribute), 280                  |
| ID (pyrogram.api.types.account.PrivacyRules attribute), 235           | ID (pyrogram.api.types.BotInlineMessageMediaContact attribute), 280               |
| ID (pyrogram.api.types.account.SentEmailCode attribute), 235          | ID (pyrogram.api.types.BotInlineMessageMediaGeo attribute), 281                   |
| ID (pyrogram.api.types.account.Takeout attribute), 236                | ID (pyrogram.api.types.BotInlineMessageMediaVenue attribute), 281                 |
| ID (pyrogram.api.types.account.TmpPassword attribute), 236            | ID (pyrogram.api.types.BotInlineMessageText attribute), 281                       |
| ID (pyrogram.api.types.account.WallPapers attribute), 236             | ID (pyrogram.api.types.BotInlineResult attribute), 282                            |
| ID (pyrogram.api.types.account.WallPapersNotModified attribute), 236  | ID (pyrogram.api.types.CdnConfig attribute), 282                                  |
| ID (pyrogram.api.types.account.WebAuthorizations attribute), 237      | ID (pyrogram.api.types.CdnPublicKey attribute), 283                               |
| ID (pyrogram.api.types.AccountDaysTTL attribute), 277                 | ID (pyrogram.api.types.Channel attribute), 283                                    |
| ID (pyrogram.api.types.auth.Authorization attribute), 237             | ID (pyrogram.api.types.ChannelAdminLogEvent attribute), 284                       |
|                                                                       | ID (pyrogram.api.types.ChannelAdminLogEventActionChangeAbout attribute), 284      |
|                                                                       | ID (pyrogram.api.types.ChannelAdminLogEventActionChangePhoto attribute), 284      |
|                                                                       | ID (pyrogram.api.types.ChannelAdminLogEventActionChangeStickerSet attribute), 284 |



- attribute), 285
- ID (pyrogram.api.types.ChannelAdminLogEventActionChangeTitle attribute), 293
- attribute), 285
- ID (pyrogram.api.types.ChannelAdminLogEventActionChangeUsername attribute), 292
- attribute), 285
- ID (pyrogram.api.types.ChannelAdminLogEventActionDefaultBannedRights attribute), 293
- attribute), 285
- ID (pyrogram.api.types.ChannelAdminLogEventActionDeleteMessage attribute), 293
- attribute), 286
- ID (pyrogram.api.types.ChannelAdminLogEventActionEditMessage attribute), 293
- attribute), 286
- ID (pyrogram.api.types.ChannelAdminLogEventActionParticipantInvite attribute), 240
- attribute), 286
- ID (pyrogram.api.types.ChannelAdminLogEventActionParticipantJoin attribute), 240
- attribute), 286
- ID (pyrogram.api.types.ChannelAdminLogEventActionParticipantLeave attribute), 241
- attribute), 287
- ID (pyrogram.api.types.ChannelAdminLogEventActionParticipantToggleAdmin attribute), 241
- attribute), 287
- ID (pyrogram.api.types.ChannelAdminLogEventActionParticipantToggleBanned attribute), 294
- attribute), 287
- ID (pyrogram.api.types.ChannelAdminLogEventActionParticipantToggleRights attribute), 294
- ID (pyrogram.api.types.ChannelAdminLogEventActionStopPoll attribute), 294
- attribute), 287
- ID (pyrogram.api.types.ChannelAdminLogEventActionTogglePrivacy attribute), 295
- attribute), 288
- ID (pyrogram.api.types.ChannelAdminLogEventActionTogglePublic attribute), 295
- ID (pyrogram.api.types.ChannelAdminLogEventActionTogglePublicHistory attribute), 296
- attribute), 288
- ID (pyrogram.api.types.ChannelAdminLogEventActionToggleSignatures attribute), 297
- attribute), 288
- ID (pyrogram.api.types.ChannelAdminLogEventActionUpdatePinned attribute), 297
- attribute), 288
- ID (pyrogram.api.types.ChannelAdminLogEventsFilter attribute), 297
- attribute), 288
- ID (pyrogram.api.types.ChannelForbidden attribute), 298
- ID (pyrogram.api.types.ChannelFull attribute), 289
- ID (pyrogram.api.types.ChannelMessagesFilter attribute), 298
- attribute), 290
- ID (pyrogram.api.types.ChannelMessagesFilterEmpty attribute), 298
- attribute), 290
- ID (pyrogram.api.types.ChannelParticipant attribute), 299
- ID (pyrogram.api.types.ChannelParticipantAdmin attribute), 299
- attribute), 291
- ID (pyrogram.api.types.ChannelParticipantBanned attribute), 299
- attribute), 291
- ID (pyrogram.api.types.ChannelParticipantCreator attribute), 299
- attribute), 292
- ID (pyrogram.api.types.ChannelParticipantsAdmins attribute), 299
- attribute), 292
- ID (pyrogram.api.types.ChannelParticipantsBanned attribute), 299
- attribute), 292
- ID (pyrogram.api.types.ChannelParticipantsBots attribute), 299
- attribute), 292
- ID (pyrogram.api.types.ChannelParticipantsContacts attribute), 299
- attribute), 293
- ID (pyrogram.api.types.ChannelParticipantSelf attribute), 299
- attribute), 292
- ID (pyrogram.api.types.ChannelParticipantsKicked attribute), 299
- attribute), 293
- ID (pyrogram.api.types.ChannelParticipantsRecent attribute), 299
- attribute), 293
- ID (pyrogram.api.types.ChannelParticipantsSearch attribute), 299
- ID (pyrogram.api.types.channels.AdminLogResults attribute), 299
- ID (pyrogram.api.types.channels.ChannelParticipant attribute), 299
- ID (pyrogram.api.types.channels.ChannelParticipants attribute), 299
- ID (pyrogram.api.types.channels.ChannelParticipantsNotModified attribute), 299
- ID (pyrogram.api.types.Chat attribute), 293
- ID (pyrogram.api.types.ChatAdminRights attribute), 294
- ID (pyrogram.api.types.ChatBannedRights attribute), 294
- ID (pyrogram.api.types.ChatEmpty attribute), 295
- ID (pyrogram.api.types.ChatForbidden attribute), 295
- ID (pyrogram.api.types.ChatFull attribute), 295
- ID (pyrogram.api.types.ChatInvite attribute), 296
- ID (pyrogram.api.types.ChatInviteAlready attribute), 296
- ID (pyrogram.api.types.ChatInviteEmpty attribute), 297
- ID (pyrogram.api.types.ChatInviteExported attribute), 297
- ID (pyrogram.api.types.ChatOnlines attribute), 297
- ID (pyrogram.api.types.ChatParticipant attribute), 297
- ID (pyrogram.api.types.ChatParticipantAdmin attribute), 298
- ID (pyrogram.api.types.ChatParticipantCreator attribute), 298
- ID (pyrogram.api.types.ChatParticipants attribute), 298
- ID (pyrogram.api.types.ChatParticipantsForbidden attribute), 298
- ID (pyrogram.api.types.ChatPhoto attribute), 299
- ID (pyrogram.api.types.ChatPhotoEmpty attribute), 299
- ID (pyrogram.api.types.ClientDHIInnerData attribute), 299
- ID (pyrogram.api.types.CodeSettings attribute), 299
- ID (pyrogram.api.types.Config attribute), 300
- ID (pyrogram.api.types.Contact attribute), 301
- ID (pyrogram.api.types.ContactBlocked attribute), 301
- ID (pyrogram.api.types.ContactLinkContact attribute), 302
- ID (pyrogram.api.types.ContactLinkHasPhone attribute), 302
- ID (pyrogram.api.types.ContactLinkNone attribute), 302
- ID (pyrogram.api.types.ContactLinkUnknown attribute), 302

- ID (`pyrogram.api.types.contacts.Blocked` attribute), 241
- ID (`pyrogram.api.types.contacts.BlockedSlice` attribute), 242
- ID (`pyrogram.api.types.contacts.Contacts` attribute), 242
- ID (`pyrogram.api.types.contacts.ContactsNotModified` attribute), 242
- ID (`pyrogram.api.types.contacts.Found` attribute), 243
- ID (`pyrogram.api.types.contacts.ImportedContacts` attribute), 243
- ID (`pyrogram.api.types.contacts.Link` attribute), 243
- ID (`pyrogram.api.types.contacts.ResolvedPeer` attribute), 244
- ID (`pyrogram.api.types.contacts.TopPeers` attribute), 244
- ID (`pyrogram.api.types.contacts.TopPeersDisabled` attribute), 244
- ID (`pyrogram.api.types.contacts.TopPeersNotModified` attribute), 245
- ID (`pyrogram.api.types.ContactStatus` attribute), 302
- ID (`pyrogram.api.types.DataJSON` attribute), 303
- ID (`pyrogram.api.types.DcOption` attribute), 303
- ID (`pyrogram.api.types.DestroyAuthKeyFail` attribute), 304
- ID (`pyrogram.api.types.DestroyAuthKeyNone` attribute), 304
- ID (`pyrogram.api.types.DestroyAuthKeyOk` attribute), 304
- ID (`pyrogram.api.types.DestroySessionNone` attribute), 304
- ID (`pyrogram.api.types.DestroySessionOk` attribute), 305
- ID (`pyrogram.api.types.DhGenFail` attribute), 305
- ID (`pyrogram.api.types.DhGenOk` attribute), 305
- ID (`pyrogram.api.types.DhGenRetry` attribute), 306
- ID (`pyrogram.api.types.Dialog` attribute), 306
- ID (`pyrogram.api.types.DialogPeer` attribute), 306
- ID (`pyrogram.api.types.Document` attribute), 307
- ID (`pyrogram.api.types.DocumentAttributeAnimated` attribute), 307
- ID (`pyrogram.api.types.DocumentAttributeAudio` attribute), 307
- ID (`pyrogram.api.types.DocumentAttributeFilename` attribute), 308
- ID (`pyrogram.api.types.DocumentAttributeHasStickers` attribute), 308
- ID (`pyrogram.api.types.DocumentAttributeImageSize` attribute), 308
- ID (`pyrogram.api.types.DocumentAttributeSticker` attribute), 308
- ID (`pyrogram.api.types.DocumentAttributeVideo` attribute), 309
- ID (`pyrogram.api.types.DocumentEmpty` attribute), 309
- ID (`pyrogram.api.types.DraftMessage` attribute), 309
- ID (`pyrogram.api.types.DraftMessageEmpty` attribute), 310
- ID (`pyrogram.api.types.EmojiKeyword` attribute), 310
- ID (`pyrogram.api.types.EmojiKeywordDeleted` attribute), 310
- ID (`pyrogram.api.types.EmojiKeywordsDifference` attribute), 310
- ID (`pyrogram.api.types.EmojiURL` attribute), 311
- ID (`pyrogram.api.types.EncryptedChat` attribute), 311
- ID (`pyrogram.api.types.EncryptedChatDiscarded` attribute), 312
- ID (`pyrogram.api.types.EncryptedChatEmpty` attribute), 312
- ID (`pyrogram.api.types.EncryptedChatRequested` attribute), 312
- ID (`pyrogram.api.types.EncryptedChatWaiting` attribute), 313
- ID (`pyrogram.api.types.EncryptedFile` attribute), 313
- ID (`pyrogram.api.types.EncryptedFileEmpty` attribute), 313
- ID (`pyrogram.api.types.EncryptedMessage` attribute), 314
- ID (`pyrogram.api.types.EncryptedMessageService` attribute), 314
- ID (`pyrogram.api.types.ExportedMessageLink` attribute), 314
- ID (`pyrogram.api.types.FileHash` attribute), 315
- ID (`pyrogram.api.types.FileLocation` attribute), 315
- ID (`pyrogram.api.types.FileLocationUnavailable` attribute), 315
- ID (`pyrogram.api.types.FoundGif` attribute), 316
- ID (`pyrogram.api.types.FoundGifCached` attribute), 316
- ID (`pyrogram.api.types.Game` attribute), 316
- ID (`pyrogram.api.types.GeoPoint` attribute), 317
- ID (`pyrogram.api.types.GeoPointEmpty` attribute), 317
- ID (`pyrogram.api.types.help.AppUpdate` attribute), 245
- ID (`pyrogram.api.types.help.ConfigSimple` attribute), 246
- ID (`pyrogram.api.types.help.DeepLinkInfo` attribute), 246
- ID (`pyrogram.api.types.help.DeepLinkInfoEmpty` attribute), 246
- ID (`pyrogram.api.types.help.InviteText` attribute), 247
- ID (`pyrogram.api.types.help.NoAppUpdate` attribute), 247
- ID (`pyrogram.api.types.help.PassportConfig` attribute), 247
- ID (`pyrogram.api.types.help.PassportConfigNotModified` attribute), 247
- ID (`pyrogram.api.types.help.ProxyDataEmpty` attribute), 248
- ID (`pyrogram.api.types.help.ProxyDataPromo` attribute), 248
- ID (`pyrogram.api.types.help.RecentMeUrls` attribute), 248
- ID (`pyrogram.api.types.help.Support` attribute), 249
- ID (`pyrogram.api.types.help.SupportName` attribute), 249

- ID (*pyrogram.api.types.help.TermsOfService* attribute), 249
- ID (*pyrogram.api.types.help.TermsOfServiceUpdate* attribute), 250
- ID (*pyrogram.api.types.help.TermsOfServiceUpdateEmpty* attribute), 250
- ID (*pyrogram.api.types.help.UserInfo* attribute), 250
- ID (*pyrogram.api.types.help.UserInfoEmpty* attribute), 251
- ID (*pyrogram.api.types.HighScore* attribute), 317
- ID (*pyrogram.api.types.HttpWait* attribute), 317
- ID (*pyrogram.api.types.ImportedContact* attribute), 318
- ID (*pyrogram.api.types.InlineBotSwitchPM* attribute), 318
- ID (*pyrogram.api.types.InputAppEvent* attribute), 318
- ID (*pyrogram.api.types.InputBotInlineMessageGame* attribute), 318
- ID (*pyrogram.api.types.InputBotInlineMessageID* attribute), 319
- ID (*pyrogram.api.types.InputBotInlineMessageMediaAuto* attribute), 319
- ID (*pyrogram.api.types.InputBotInlineMessageMediaContact* attribute), 319
- ID (*pyrogram.api.types.InputBotInlineMessageMediaGeo* attribute), 320
- ID (*pyrogram.api.types.InputBotInlineMessageMediaVenue* attribute), 320
- ID (*pyrogram.api.types.InputBotInlineMessageText* attribute), 321
- ID (*pyrogram.api.types.InputBotInlineResult* attribute), 321
- ID (*pyrogram.api.types.InputBotInlineResultDocument* attribute), 321
- ID (*pyrogram.api.types.InputBotInlineResultGame* attribute), 322
- ID (*pyrogram.api.types.InputBotInlineResultPhoto* attribute), 322
- ID (*pyrogram.api.types.InputChannel* attribute), 323
- ID (*pyrogram.api.types.InputChannelEmpty* attribute), 323
- ID (*pyrogram.api.types.InputChatPhoto* attribute), 323
- ID (*pyrogram.api.types.InputChatPhotoEmpty* attribute), 323
- ID (*pyrogram.api.types.InputChatUploadedPhoto* attribute), 323
- ID (*pyrogram.api.types.InputCheckPasswordEmpty* attribute), 324
- ID (*pyrogram.api.types.InputCheckPasswordSRP* attribute), 324
- ID (*pyrogram.api.types.InputClientProxy* attribute), 324
- ID (*pyrogram.api.types.InputDialogPeer* attribute), 324
- ID (*pyrogram.api.types.InputDocument* attribute), 325
- ID (*pyrogram.api.types.InputDocumentEmpty* attribute), 325
- ID (*pyrogram.api.types.InputDocumentFileLocation* attribute), 325
- ID (*pyrogram.api.types.InputEncryptedChat* attribute), 325
- ID (*pyrogram.api.types.InputEncryptedFile* attribute), 326
- ID (*pyrogram.api.types.InputEncryptedFileBigUploaded* attribute), 326
- ID (*pyrogram.api.types.InputEncryptedFileEmpty* attribute), 326
- ID (*pyrogram.api.types.InputEncryptedFileLocation* attribute), 326
- ID (*pyrogram.api.types.InputEncryptedFileUploaded* attribute), 327
- ID (*pyrogram.api.types.InputFile* attribute), 327
- ID (*pyrogram.api.types.InputFileBig* attribute), 327
- ID (*pyrogram.api.types.InputFileLocation* attribute), 327
- ID (*pyrogram.api.types.InputGameID* attribute), 328
- ID (*pyrogram.api.types.InputGameShortName* attribute), 328
- ID (*pyrogram.api.types.InputGeoPoint* attribute), 328
- ID (*pyrogram.api.types.InputGeoPointEmpty* attribute), 329
- ID (*pyrogram.api.types.InputMediaContact* attribute), 329
- ID (*pyrogram.api.types.InputMediaDocument* attribute), 329
- ID (*pyrogram.api.types.InputMediaDocumentExternal* attribute), 329
- ID (*pyrogram.api.types.InputMediaEmpty* attribute), 330
- ID (*pyrogram.api.types.InputMediaGame* attribute), 330
- ID (*pyrogram.api.types.InputMediaGeoLive* attribute), 330
- ID (*pyrogram.api.types.InputMediaGeoPoint* attribute), 330
- ID (*pyrogram.api.types.InputMediaGifExternal* attribute), 330
- ID (*pyrogram.api.types.InputMediaInvoice* attribute), 331
- ID (*pyrogram.api.types.InputMediaPhoto* attribute), 331
- ID (*pyrogram.api.types.InputMediaPhotoExternal* attribute), 331
- ID (*pyrogram.api.types.InputMediaPoll* attribute), 332
- ID (*pyrogram.api.types.InputMediaUploadedDocument* attribute), 332
- ID (*pyrogram.api.types.InputMediaUploadedPhoto* attribute), 332
- ID (*pyrogram.api.types.InputMediaVenue* attribute), 332
- ID (*pyrogram.api.types.InputMessageEntityMentionName* attribute), 333
- ID (*pyrogram.api.types.InputMessageID* attribute), 333
- ID (*pyrogram.api.types.InputMessagePinned* attribute), 333
- ID (*pyrogram.api.types.InputMessageReplyTo* attribute), 333

- 334
- ID (*pyrogram.api.types.InputMessagesFilterChatPhotos* attribute), 334
- ID (*pyrogram.api.types.InputMessagesFilterContacts* attribute), 334
- ID (*pyrogram.api.types.InputMessagesFilterDocument* attribute), 334
- ID (*pyrogram.api.types.InputMessagesFilterEmpty* attribute), 334
- ID (*pyrogram.api.types.InputMessagesFilterGeo* attribute), 335
- ID (*pyrogram.api.types.InputMessagesFilterGif* attribute), 335
- ID (*pyrogram.api.types.InputMessagesFilterMusic* attribute), 335
- ID (*pyrogram.api.types.InputMessagesFilterMyMentions* attribute), 335
- ID (*pyrogram.api.types.InputMessagesFilterPhoneCalls* attribute), 335
- ID (*pyrogram.api.types.InputMessagesFilterPhotos* attribute), 336
- ID (*pyrogram.api.types.InputMessagesFilterPhotoVideo* attribute), 336
- ID (*pyrogram.api.types.InputMessagesFilterRoundVideo* attribute), 336
- ID (*pyrogram.api.types.InputMessagesFilterRoundVoice* attribute), 336
- ID (*pyrogram.api.types.InputMessagesFilterUrl* attribute), 336
- ID (*pyrogram.api.types.InputMessagesFilterVideo* attribute), 337
- ID (*pyrogram.api.types.InputMessagesFilterVoice* attribute), 337
- ID (*pyrogram.api.types.InputNotifyBroadcasts* attribute), 337
- ID (*pyrogram.api.types.InputNotifyChats* attribute), 337
- ID (*pyrogram.api.types.InputNotifyPeer* attribute), 337
- ID (*pyrogram.api.types.InputNotifyUsers* attribute), 338
- ID (*pyrogram.api.types.InputPaymentCredentials* attribute), 338
- ID (*pyrogram.api.types.InputPaymentCredentialsAndroidPay* attribute), 338
- ID (*pyrogram.api.types.InputPaymentCredentialsApplePay* attribute), 338
- ID (*pyrogram.api.types.InputPaymentCredentialsSaved* attribute), 338
- ID (*pyrogram.api.types.InputPeerChannel* attribute), 339
- ID (*pyrogram.api.types.InputPeerChat* attribute), 339
- ID (*pyrogram.api.types.InputPeerEmpty* attribute), 339
- ID (*pyrogram.api.types.InputPeerNotifySettings* attribute), 339
- ID (*pyrogram.api.types.InputPeerSelf* attribute), 340
- ID (*pyrogram.api.types.InputPeerUser* attribute), 340
- ID (*pyrogram.api.types.InputPhoneCall* attribute), 340
- ID (*pyrogram.api.types.InputPhoneContact* attribute), 340
- ID (*pyrogram.api.types.InputPhoto* attribute), 341
- ID (*pyrogram.api.types.InputPhotoEmpty* attribute), 341
- ID (*pyrogram.api.types.InputPrivacyKeyChatInvite* attribute), 341
- ID (*pyrogram.api.types.InputPrivacyKeyForwards* attribute), 341
- ID (*pyrogram.api.types.InputPrivacyKeyPhoneCall* attribute), 341
- ID (*pyrogram.api.types.InputPrivacyKeyPhoneP2P* attribute), 342
- ID (*pyrogram.api.types.InputPrivacyKeyProfilePhoto* attribute), 342
- ID (*pyrogram.api.types.InputPrivacyKeyStatusTimestamp* attribute), 342
- ID (*pyrogram.api.types.InputPrivacyValueAllowAll* attribute), 342
- ID (*pyrogram.api.types.InputPrivacyValueAllowContacts* attribute), 342
- ID (*pyrogram.api.types.InputPrivacyValueAllowUsers* attribute), 343
- ID (*pyrogram.api.types.InputPrivacyValueDisallowAll* attribute), 343
- ID (*pyrogram.api.types.InputPrivacyValueDisallowContacts* attribute), 343
- ID (*pyrogram.api.types.InputPrivacyValueDisallowUsers* attribute), 343
- ID (*pyrogram.api.types.InputReportReasonChildAbuse* attribute), 343
- ID (*pyrogram.api.types.InputReportReasonCopyright* attribute), 344
- ID (*pyrogram.api.types.InputReportReasonOther* attribute), 344
- ID (*pyrogram.api.types.InputReportReasonPornography* attribute), 344
- ID (*pyrogram.api.types.InputReportReasonSpam* attribute), 344
- ID (*pyrogram.api.types.InputReportReasonViolence* attribute), 344
- ID (*pyrogram.api.types.InputSecureFile* attribute), 345
- ID (*pyrogram.api.types.InputSecureFileLocation* attribute), 345
- ID (*pyrogram.api.types.InputSecureFileUploaded* attribute), 345
- ID (*pyrogram.api.types.InputSecureValue* attribute), 345
- ID (*pyrogram.api.types.InputSingleMedia* attribute), 346
- ID (*pyrogram.api.types.InputStickeredMediaDocument* attribute), 347
- ID (*pyrogram.api.types.InputStickeredMediaPhoto* attribute), 348
- ID (*pyrogram.api.types.InputStickerSetEmpty* attribute), 347
- ID (*pyrogram.api.types.InputStickerSetID* attribute), 347



- ID (*pyrogram.api.types.InputStickerSetItem* attribute), 347
- ID (*pyrogram.api.types.InputStickerSetShortName* attribute), 347
- ID (*pyrogram.api.types.InputTakeoutFileLocation* attribute), 348
- ID (*pyrogram.api.types.InputUser* attribute), 348
- ID (*pyrogram.api.types.InputUserEmpty* attribute), 348
- ID (*pyrogram.api.types.InputUserSelf* attribute), 349
- ID (*pyrogram.api.types.InputWallPaper* attribute), 349
- ID (*pyrogram.api.types.InputWallPaperSlug* attribute), 349
- ID (*pyrogram.api.types.InputWebDocument* attribute), 349
- ID (*pyrogram.api.types.InputWebFileGeoPointLocation* attribute), 350
- ID (*pyrogram.api.types.InputWebFileLocation* attribute), 350
- ID (*pyrogram.api.types.Invoice* attribute), 350
- ID (*pyrogram.api.types.IpPort* attribute), 351
- ID (*pyrogram.api.types.IpPortSecret* attribute), 351
- ID (*pyrogram.api.types.JsonArray* attribute), 351
- ID (*pyrogram.api.types.JsonBool* attribute), 351
- ID (*pyrogram.api.types.JsonNull* attribute), 352
- ID (*pyrogram.api.types.JsonNumber* attribute), 352
- ID (*pyrogram.api.types.JsonObject* attribute), 352
- ID (*pyrogram.api.types.JsonObjectValue* attribute), 352
- ID (*pyrogram.api.types.JsonString* attribute), 353
- ID (*pyrogram.api.types.KeyboardButton* attribute), 353
- ID (*pyrogram.api.types.KeyboardButtonBuy* attribute), 353
- ID (*pyrogram.api.types.KeyboardButtonCallback* attribute), 353
- ID (*pyrogram.api.types.KeyboardButtonGame* attribute), 354
- ID (*pyrogram.api.types.KeyboardButtonRequestGeoLocation* attribute), 354
- ID (*pyrogram.api.types.KeyboardButtonRequestPhone* attribute), 354
- ID (*pyrogram.api.types.KeyboardButtonRow* attribute), 354
- ID (*pyrogram.api.types.KeyboardButtonSwitchInline* attribute), 354
- ID (*pyrogram.api.types.KeyboardButtonUrl* attribute), 355
- ID (*pyrogram.api.types.LabeledPrice* attribute), 355
- ID (*pyrogram.api.types.LangPackDifference* attribute), 355
- ID (*pyrogram.api.types.LangPackLanguage* attribute), 356
- ID (*pyrogram.api.types.LangPackString* attribute), 356
- ID (*pyrogram.api.types.LangPackStringDeleted* attribute), 356
- ID (*pyrogram.api.types.LangPackStringPluralized* attribute), 357
- ID (*pyrogram.api.types.MaskCoords* attribute), 357
- ID (*pyrogram.api.types.Message* attribute), 357
- ID (*pyrogram.api.types.MessageActionBotAllowed* attribute), 358
- ID (*pyrogram.api.types.MessageActionChannelCreate* attribute), 359
- ID (*pyrogram.api.types.MessageActionChannelMigrateFrom* attribute), 359
- ID (*pyrogram.api.types.MessageActionChatAddUser* attribute), 359
- ID (*pyrogram.api.types.MessageActionChatCreate* attribute), 359
- ID (*pyrogram.api.types.MessageActionChatDeletePhoto* attribute), 360
- ID (*pyrogram.api.types.MessageActionChatDeleteUser* attribute), 360
- ID (*pyrogram.api.types.MessageActionChatEditPhoto* attribute), 360
- ID (*pyrogram.api.types.MessageActionChatEditTitle* attribute), 360
- ID (*pyrogram.api.types.MessageActionChatJoinedByLink* attribute), 360
- ID (*pyrogram.api.types.MessageActionChatMigrateTo* attribute), 361
- ID (*pyrogram.api.types.MessageActionContactSignUp* attribute), 361
- ID (*pyrogram.api.types.MessageActionCustomAction* attribute), 361
- ID (*pyrogram.api.types.MessageActionEmpty* attribute), 361
- ID (*pyrogram.api.types.MessageActionGameScore* attribute), 361
- ID (*pyrogram.api.types.MessageActionHistoryClear* attribute), 362
- ID (*pyrogram.api.types.MessageActionPaymentSent* attribute), 362
- ID (*pyrogram.api.types.MessageActionPaymentSentMe* attribute), 362
- ID (*pyrogram.api.types.MessageActionPhoneCall* attribute), 362
- ID (*pyrogram.api.types.MessageActionPinMessage* attribute), 363
- ID (*pyrogram.api.types.MessageActionScreenshotTaken* attribute), 363
- ID (*pyrogram.api.types.MessageActionSecureValuesSent* attribute), 363
- ID (*pyrogram.api.types.MessageActionSecureValuesSentMe* attribute), 363
- ID (*pyrogram.api.types.MessageEmpty* attribute), 364
- ID (*pyrogram.api.types.MessageEntityBold* attribute), 364
- ID (*pyrogram.api.types.MessageEntityBotCommand* attribute), 364

- ID ([pyrogram.api.types.MessageEntityCashtag](#) attribute), 364
- ID ([pyrogram.api.types.MessageEntityCode](#) attribute), 365
- ID ([pyrogram.api.types.MessageEntityEmail](#) attribute), 365
- ID ([pyrogram.api.types.MessageEntityHashtag](#) attribute), 365
- ID ([pyrogram.api.types.MessageEntityItalic](#) attribute), 365
- ID ([pyrogram.api.types.MessageEntityMention](#) attribute), 366
- ID ([pyrogram.api.types.MessageEntityMentionName](#) attribute), 366
- ID ([pyrogram.api.types.MessageEntityPhone](#) attribute), 366
- ID ([pyrogram.api.types.MessageEntityPre](#) attribute), 366
- ID ([pyrogram.api.types.MessageEntityTextUrl](#) attribute), 367
- ID ([pyrogram.api.types.MessageEntityUnknown](#) attribute), 367
- ID ([pyrogram.api.types.MessageEntityUrl](#) attribute), 367
- ID ([pyrogram.api.types.MessageFwdHeader](#) attribute), 367
- ID ([pyrogram.api.types.MessageMediaContact](#) attribute), 368
- ID ([pyrogram.api.types.MessageMediaDocument](#) attribute), 368
- ID ([pyrogram.api.types.MessageMediaEmpty](#) attribute), 369
- ID ([pyrogram.api.types.MessageMediaGame](#) attribute), 369
- ID ([pyrogram.api.types.MessageMediaGeo](#) attribute), 369
- ID ([pyrogram.api.types.MessageMediaGeoLive](#) attribute), 369
- ID ([pyrogram.api.types.MessageMediaInvoice](#) attribute), 370
- ID ([pyrogram.api.types.MessageMediaPhoto](#) attribute), 370
- ID ([pyrogram.api.types.MessageMediaPoll](#) attribute), 371
- ID ([pyrogram.api.types.MessageMediaUnsupported](#) attribute), 371
- ID ([pyrogram.api.types.MessageMediaVenue](#) attribute), 371
- ID ([pyrogram.api.types.MessageMediaWebPage](#) attribute), 372
- ID ([pyrogram.api.types.MessageRange](#) attribute), 372
- ID ([pyrogram.api.types.messages.AffectedHistory](#) attribute), 251
- ID ([pyrogram.api.types.messages.AffectedMessages](#) attribute), 251
- ID ([pyrogram.api.types.messages.AllStickers](#) attribute), 252
- ID ([pyrogram.api.types.messages.AllStickersNotModified](#) attribute), 252
- ID ([pyrogram.api.types.messages.ArchivedStickers](#) attribute), 252
- ID ([pyrogram.api.types.messages.BotCallbackAnswer](#) attribute), 252
- ID ([pyrogram.api.types.messages.BotResults](#) attribute), 253
- ID ([pyrogram.api.types.messages.ChannelMessages](#) attribute), 253
- ID ([pyrogram.api.types.messages.ChatFull](#) attribute), 254
- ID ([pyrogram.api.types.messages.Chats](#) attribute), 254
- ID ([pyrogram.api.types.messages.ChatsSlice](#) attribute), 254
- ID ([pyrogram.api.types.messages.DhConfig](#) attribute), 255
- ID ([pyrogram.api.types.messages.DhConfigNotModified](#) attribute), 255
- ID ([pyrogram.api.types.messages.Dialogs](#) attribute), 256
- ID ([pyrogram.api.types.messages.DialogsNotModified](#) attribute), 256
- ID ([pyrogram.api.types.messages.DialogsSlice](#) attribute), 256
- ID ([pyrogram.api.types.messages.FavedStickers](#) attribute), 257
- ID ([pyrogram.api.types.messages.FavedStickersNotModified](#) attribute), 257
- ID ([pyrogram.api.types.messages.FeaturedStickers](#) attribute), 257
- ID ([pyrogram.api.types.messages.FeaturedStickersNotModified](#) attribute), 258
- ID ([pyrogram.api.types.messages.FoundGifs](#) attribute), 258
- ID ([pyrogram.api.types.messages.FoundStickerSets](#) attribute), 258
- ID ([pyrogram.api.types.messages.FoundStickerSetsNotModified](#) attribute), 258
- ID ([pyrogram.api.types.messages.HighScores](#) attribute), 259
- ID ([pyrogram.api.types.messages.MessageEditData](#) attribute), 259
- ID ([pyrogram.api.types.messages.Messages](#) attribute), 259
- ID ([pyrogram.api.types.messages.MessagesNotModified](#) attribute), 260
- ID ([pyrogram.api.types.messages.MessagesSlice](#) attribute), 260
- ID ([pyrogram.api.types.messages.PeerDialogs](#) attribute), 260
- ID ([pyrogram.api.types.messages.RecentStickers](#) attribute), 261
- ID ([pyrogram.api.types.messages.RecentStickersNotModified](#) attribute), 261

- attribute*), 261
- ID (*pyrogram.api.types.messages.SavedGifs attribute*), 261
- ID (*pyrogram.api.types.messages.SavedGifsNotModified attribute*), 262
- ID (*pyrogram.api.types.messages.SentEncryptedFile attribute*), 262
- ID (*pyrogram.api.types.messages.SentEncryptedMessage attribute*), 262
- ID (*pyrogram.api.types.messages.Stickers attribute*), 263
- ID (*pyrogram.api.types.messages.StickerSet attribute*), 263
- ID (*pyrogram.api.types.messages.StickerSetInstallResultArchive attribute*), 263
- ID (*pyrogram.api.types.messages.StickerSetInstallResultSuccess attribute*), 263
- ID (*pyrogram.api.types.messages.StickersNotModified attribute*), 264
- ID (*pyrogram.api.types.MessageService attribute*), 372
- ID (*pyrogram.api.types.MsgDetailedInfo attribute*), 373
- ID (*pyrogram.api.types.MsgNewDetailedInfo attribute*), 373
- ID (*pyrogram.api.types.MsgResendAnsReq attribute*), 373
- ID (*pyrogram.api.types.MsgResendReq attribute*), 374
- ID (*pyrogram.api.types.MsgsAck attribute*), 374
- ID (*pyrogram.api.types.MsgsAllInfo attribute*), 374
- ID (*pyrogram.api.types.MsgsStateInfo attribute*), 374
- ID (*pyrogram.api.types.MsgsStateReq attribute*), 375
- ID (*pyrogram.api.types.NearestDc attribute*), 375
- ID (*pyrogram.api.types.NewSessionCreated attribute*), 375
- ID (*pyrogram.api.types.NotifyBroadcasts attribute*), 375
- ID (*pyrogram.api.types.NotifyChats attribute*), 376
- ID (*pyrogram.api.types.NotifyPeer attribute*), 376
- ID (*pyrogram.api.types.NotifyUsers attribute*), 376
- ID (*pyrogram.api.types.Page attribute*), 378
- ID (*pyrogram.api.types.PageBlockAnchor attribute*), 378
- ID (*pyrogram.api.types.PageBlockAudio attribute*), 379
- ID (*pyrogram.api.types.PageBlockAuthorDate attribute*), 379
- ID (*pyrogram.api.types.PageBlockBlockquote attribute*), 379
- ID (*pyrogram.api.types.PageBlockChannel attribute*), 379
- ID (*pyrogram.api.types.PageBlockCollage attribute*), 380
- ID (*pyrogram.api.types.PageBlockCover attribute*), 380
- ID (*pyrogram.api.types.PageBlockDetails attribute*), 380
- ID (*pyrogram.api.types.PageBlockDivider attribute*), 381
- ID (*pyrogram.api.types.PageBlockEmbed attribute*), 381
- ID (*pyrogram.api.types.PageBlockEmbedPost attribute*), 382
- ID (*pyrogram.api.types.PageBlockFooter attribute*), 382
- ID (*pyrogram.api.types.PageBlockHeader attribute*), 382
- ID (*pyrogram.api.types.PageBlockKicker attribute*), 383
- ID (*pyrogram.api.types.PageBlockList attribute*), 383
- ID (*pyrogram.api.types.PageBlockMap attribute*), 383
- ID (*pyrogram.api.types.PageBlockOrderedList attribute*), 383
- ID (*pyrogram.api.types.PageBlockParagraph attribute*), 384
- ID (*pyrogram.api.types.PageBlockPhoto attribute*), 384
- ID (*pyrogram.api.types.PageBlockPreformatted attribute*), 384
- ID (*pyrogram.api.types.PageBlockPullquote attribute*), 385
- ID (*pyrogram.api.types.PageBlockRelatedArticles attribute*), 385
- ID (*pyrogram.api.types.PageBlockSlideshow attribute*), 385
- ID (*pyrogram.api.types.PageBlockSubheader attribute*), 386
- ID (*pyrogram.api.types.PageBlockSubtitle attribute*), 386
- ID (*pyrogram.api.types.PageBlockTable attribute*), 386
- ID (*pyrogram.api.types.PageBlockTitle attribute*), 387
- ID (*pyrogram.api.types.PageBlockUnsupported attribute*), 387
- ID (*pyrogram.api.types.PageBlockVideo attribute*), 387
- ID (*pyrogram.api.types.PageCaption attribute*), 387
- ID (*pyrogram.api.types.PageListItemBlocks attribute*), 388
- ID (*pyrogram.api.types.PageListItemText attribute*), 388
- ID (*pyrogram.api.types.PageListOrderedItemBlocks attribute*), 388
- ID (*pyrogram.api.types.PageListOrderedItemText attribute*), 389
- ID (*pyrogram.api.types.PageRelatedArticle attribute*), 389
- ID (*pyrogram.api.types.PageTableCell attribute*), 389
- ID (*pyrogram.api.types.PageTableRow attribute*), 390
- ID (*pyrogram.api.types.PasswordKdfAlgoSHA256SHA256PBKDF2HMAC attribute*), 390
- ID (*pyrogram.api.types.PasswordKdfAlgoUnknown attribute*), 390
- ID (*pyrogram.api.types.PaymentCharge attribute*), 391
- ID (*pyrogram.api.types.PaymentRequestedInfo attribute*), 391
- ID (*pyrogram.api.types.payments.PaymentForm attribute*), 264
- ID (*pyrogram.api.types.payments.PaymentReceipt attribute*), 265
- ID (*pyrogram.api.types.payments.PaymentResult attribute*), 265
- ID (*pyrogram.api.types.payments.PaymentVerificationNeeded attribute*), 265
- ID (*pyrogram.api.types.payments.SavedInfo attribute*), 266
- ID (*pyrogram.api.types.payments.ValidatedRequestedInfo*

- attribute*), 266
- ID (*pyrogram.api.types.PaymentSavedCredentialsCard attribute*), 391
- ID (*pyrogram.api.types.PeerChannel attribute*), 391
- ID (*pyrogram.api.types.PeerChat attribute*), 392
- ID (*pyrogram.api.types.PeerNotifySettings attribute*), 392
- ID (*pyrogram.api.types.PeerSettings attribute*), 392
- ID (*pyrogram.api.types.PeerUser attribute*), 392
- ID (*pyrogram.api.types.phone.PhoneCall attribute*), 266
- ID (*pyrogram.api.types.PhoneCall attribute*), 393
- ID (*pyrogram.api.types.PhoneCallAccepted attribute*), 393
- ID (*pyrogram.api.types.PhoneCallDiscarded attribute*), 394
- ID (*pyrogram.api.types.PhoneCallDiscardReasonBusy attribute*), 394
- ID (*pyrogram.api.types.PhoneCallDiscardReasonDisconnect attribute*), 394
- ID (*pyrogram.api.types.PhoneCallDiscardReasonHangup attribute*), 394
- ID (*pyrogram.api.types.PhoneCallDiscardReasonMissed attribute*), 394
- ID (*pyrogram.api.types.PhoneCallEmpty attribute*), 395
- ID (*pyrogram.api.types.PhoneCallProtocol attribute*), 395
- ID (*pyrogram.api.types.PhoneCallRequested attribute*), 395
- ID (*pyrogram.api.types.PhoneCallWaiting attribute*), 396
- ID (*pyrogram.api.types.PhoneConnection attribute*), 396
- ID (*pyrogram.api.types.Photo attribute*), 396
- ID (*pyrogram.api.types.PhotoCachedSize attribute*), 397
- ID (*pyrogram.api.types.PhotoEmpty attribute*), 397
- ID (*pyrogram.api.types.photos.Photo attribute*), 267
- ID (*pyrogram.api.types.photos.Photos attribute*), 267
- ID (*pyrogram.api.types.photos.PhotosSlice attribute*), 267
- ID (*pyrogram.api.types.PhotoSize attribute*), 397
- ID (*pyrogram.api.types.PhotoSizeEmpty attribute*), 398
- ID (*pyrogram.api.types.PhotoStrippedSize attribute*), 398
- ID (*pyrogram.api.types.Poll attribute*), 398
- ID (*pyrogram.api.types.PollAnswer attribute*), 398
- ID (*pyrogram.api.types.PollAnswerVoters attribute*), 399
- ID (*pyrogram.api.types.PollResults attribute*), 399
- ID (*pyrogram.api.types.Pong attribute*), 399
- ID (*pyrogram.api.types.PopularContact attribute*), 399
- ID (*pyrogram.api.types.PostAddress attribute*), 400
- ID (*pyrogram.api.types.PQInnerData attribute*), 376
- ID (*pyrogram.api.types.PQInnerDataDc attribute*), 377
- ID (*pyrogram.api.types.PQInnerDataTemp attribute*), 377
- ID (*pyrogram.api.types.PQInnerDataTempDc attribute*), 377
- ID (*pyrogram.api.types.PrivacyKeyChatInvite attribute*), 400
- ID (*pyrogram.api.types.PrivacyKeyForwards attribute*), 400
- ID (*pyrogram.api.types.PrivacyKeyPhoneCall attribute*), 400
- ID (*pyrogram.api.types.PrivacyKeyPhoneP2P attribute*), 401
- ID (*pyrogram.api.types.PrivacyKeyProfilePhoto attribute*), 401
- ID (*pyrogram.api.types.PrivacyKeyStatusTimestamp attribute*), 401
- ID (*pyrogram.api.types.PrivacyValueAllowAll attribute*), 401
- ID (*pyrogram.api.types.PrivacyValueAllowContacts attribute*), 401
- ID (*pyrogram.api.types.PrivacyValueAllowUsers attribute*), 402
- ID (*pyrogram.api.types.PrivacyValueDisallowAll attribute*), 402
- ID (*pyrogram.api.types.PrivacyValueDisallowContacts attribute*), 402
- ID (*pyrogram.api.types.PrivacyValueDisallowUsers attribute*), 402
- ID (*pyrogram.api.types.ReceivedNotifyMessage attribute*), 402
- ID (*pyrogram.api.types.RecentMeUrlChat attribute*), 403
- ID (*pyrogram.api.types.RecentMeUrlChatInvite attribute*), 403
- ID (*pyrogram.api.types.RecentMeUrlStickerSet attribute*), 403
- ID (*pyrogram.api.types.RecentMeUrlUnknown attribute*), 403
- ID (*pyrogram.api.types.RecentMeUrlUser attribute*), 404
- ID (*pyrogram.api.types.ReplyInlineMarkup attribute*), 404
- ID (*pyrogram.api.types.ReplyKeyboardForceReply attribute*), 404
- ID (*pyrogram.api.types.ReplyKeyboardHide attribute*), 404
- ID (*pyrogram.api.types.ReplyKeyboardMarkup attribute*), 404
- ID (*pyrogram.api.types.ResPQ attribute*), 405
- ID (*pyrogram.api.types.RpcAnswerDropped attribute*), 405
- ID (*pyrogram.api.types.RpcAnswerDroppedRunning attribute*), 405
- ID (*pyrogram.api.types.RpcAnswerUnknown attribute*), 406
- ID (*pyrogram.api.types.RpcError attribute*), 406
- ID (*pyrogram.api.types.RpcResult attribute*), 406
- ID (*pyrogram.api.types.SavedPhoneContact attribute*), 406
- ID (*pyrogram.api.types.SecureCredentialsEncrypted attribute*), 406



- [tribute\), 407](#)
- ID ([pyrogram.api.types.SecureData attribute](#)), [407](#)
- ID ([pyrogram.api.types.SecureFile attribute](#)), [407](#)
- ID ([pyrogram.api.types.SecureFileEmpty attribute](#)), [408](#)
- ID ([pyrogram.api.types.SecurePasswordKdfAlgoPBKDF2HMACSHA512 attribute](#)), [408](#)
- ID ([pyrogram.api.types.SecurePasswordKdfAlgoSHA512 attribute](#)), [408](#)
- ID ([pyrogram.api.types.SecurePasswordKdfAlgoUnknown attribute](#)), [408](#)
- ID ([pyrogram.api.types.SecurePlainEmail attribute](#)), [408](#)
- ID ([pyrogram.api.types.SecurePlainPhone attribute](#)), [409](#)
- ID ([pyrogram.api.types.SecureRequiredType attribute](#)), [409](#)
- ID ([pyrogram.api.types.SecureRequiredTypeOneOf attribute](#)), [409](#)
- ID ([pyrogram.api.types.SecureSecretSettings attribute](#)), [409](#)
- ID ([pyrogram.api.types.SecureValue attribute](#)), [410](#)
- ID ([pyrogram.api.types.SecureValueError attribute](#)), [410](#)
- ID ([pyrogram.api.types.SecureValueErrorData attribute](#)), [411](#)
- ID ([pyrogram.api.types.SecureValueErrorFile attribute](#)), [411](#)
- ID ([pyrogram.api.types.SecureValueErrorFiles attribute](#)), [412](#)
- ID ([pyrogram.api.types.SecureValueErrorFrontSide attribute](#)), [412](#)
- ID ([pyrogram.api.types.SecureValueErrorReverseSide attribute](#)), [412](#)
- ID ([pyrogram.api.types.SecureValueErrorSelfie attribute](#)), [413](#)
- ID ([pyrogram.api.types.SecureValueErrorTranslationFile attribute](#)), [413](#)
- ID ([pyrogram.api.types.SecureValueErrorTranslationFiles attribute](#)), [414](#)
- ID ([pyrogram.api.types.SecureValueHash attribute](#)), [414](#)
- ID ([pyrogram.api.types.SecureValueTypeAddress attribute](#)), [414](#)
- ID ([pyrogram.api.types.SecureValueTypeBankStatement attribute](#)), [414](#)
- ID ([pyrogram.api.types.SecureValueTypeDriverLicense attribute](#)), [415](#)
- ID ([pyrogram.api.types.SecureValueTypeEmail attribute](#)), [415](#)
- ID ([pyrogram.api.types.SecureValueTypeIdentityCard attribute](#)), [415](#)
- ID ([pyrogram.api.types.SecureValueTypeInternalPassport attribute](#)), [415](#)
- ID ([pyrogram.api.types.SecureValueTypePassport attribute](#)), [415](#)
- ID ([pyrogram.api.types.SecureValueTypePassportRegistration attribute](#)), [416](#)
- ID ([pyrogram.api.types.SecureValueTypePersonalDetails attribute](#)), [416](#)
- ID ([pyrogram.api.types.SecureValueTypePhone attribute](#)), [416](#)
- ID ([pyrogram.api.types.SecureValueTypeRentalAgreement attribute](#)), [416](#)
- ID ([pyrogram.api.types.SecureValueTypeTemporaryRegistration attribute](#)), [416](#)
- ID ([pyrogram.api.types.SecureValueTypeUtilityBill attribute](#)), [417](#)
- ID ([pyrogram.api.types.SendMessageCancelAction attribute](#)), [417](#)
- ID ([pyrogram.api.types.SendMessageChooseContactAction attribute](#)), [417](#)
- ID ([pyrogram.api.types.SendMessageGamePlayAction attribute](#)), [417](#)
- ID ([pyrogram.api.types.SendMessageGeoLocationAction attribute](#)), [417](#)
- ID ([pyrogram.api.types.SendMessageRecordAudioAction attribute](#)), [418](#)
- ID ([pyrogram.api.types.SendMessageRecordRoundAction attribute](#)), [418](#)
- ID ([pyrogram.api.types.SendMessageRecordVideoAction attribute](#)), [418](#)
- ID ([pyrogram.api.types.SendMessageTypingAction attribute](#)), [418](#)
- ID ([pyrogram.api.types.SendMessageUploadAudioAction attribute](#)), [418](#)
- ID ([pyrogram.api.types.SendMessageUploadDocumentAction attribute](#)), [419](#)
- ID ([pyrogram.api.types.SendMessageUploadPhotoAction attribute](#)), [419](#)
- ID ([pyrogram.api.types.SendMessageUploadRoundAction attribute](#)), [419](#)
- ID ([pyrogram.api.types.SendMessageUploadVideoAction attribute](#)), [419](#)
- ID ([pyrogram.api.types.ServerDHInnerData attribute](#)), [419](#)
- ID ([pyrogram.api.types.ServerDHParamsFail attribute](#)), [420](#)
- ID ([pyrogram.api.types.ServerDHParamsOk attribute](#)), [420](#)
- ID ([pyrogram.api.types.ShippingOption attribute](#)), [420](#)
- ID ([pyrogram.api.types.StatsURL attribute](#)), [421](#)
- ID ([pyrogram.api.types.StickerPack attribute](#)), [421](#)
- ID ([pyrogram.api.types.StickerSet attribute](#)), [421](#)
- ID ([pyrogram.api.types.StickerSetCovered attribute](#)), [422](#)
- ID ([pyrogram.api.types.StickerSetMultiCovered attribute](#)), [422](#)
- ID ([pyrogram.api.types.storage.FileGif attribute](#)), [268](#)
- ID ([pyrogram.api.types.storage.FileJpeg attribute](#)), [268](#)
- ID ([pyrogram.api.types.storage.FileMov attribute](#)), [268](#)
- ID ([pyrogram.api.types.storage.FileMp3 attribute](#)), [268](#)
- ID ([pyrogram.api.types.storage.FileMp4 attribute](#)), [268](#)
- ID ([pyrogram.api.types.storage.FilePartial attribute](#)), [268](#)

- 269
- ID (*pyrogram.api.types.storage.FilePdf* attribute), 269
- ID (*pyrogram.api.types.storage.FilePng* attribute), 269
- ID (*pyrogram.api.types.storage.FileUnknown* attribute), 269
- ID (*pyrogram.api.types.storage.FileWebp* attribute), 269
- ID (*pyrogram.api.types.TextAnchor* attribute), 422
- ID (*pyrogram.api.types.TextBold* attribute), 423
- ID (*pyrogram.api.types.TextConcat* attribute), 423
- ID (*pyrogram.api.types.TextEmail* attribute), 423
- ID (*pyrogram.api.types.TextEmpty* attribute), 423
- ID (*pyrogram.api.types.TextFixed* attribute), 424
- ID (*pyrogram.api.types.TextImage* attribute), 424
- ID (*pyrogram.api.types.TextItalic* attribute), 424
- ID (*pyrogram.api.types.TextMarked* attribute), 424
- ID (*pyrogram.api.types.TextPhone* attribute), 425
- ID (*pyrogram.api.types.TextPlain* attribute), 425
- ID (*pyrogram.api.types.TextStrike* attribute), 425
- ID (*pyrogram.api.types.TextSubscript* attribute), 425
- ID (*pyrogram.api.types.TextSuperscript* attribute), 426
- ID (*pyrogram.api.types.TextUnderline* attribute), 426
- ID (*pyrogram.api.types.TextUrl* attribute), 426
- ID (*pyrogram.api.types.TopPeer* attribute), 426
- ID (*pyrogram.api.types.TopPeerCategoryBotsInline* attribute), 427
- ID (*pyrogram.api.types.TopPeerCategoryBotsPM* attribute), 427
- ID (*pyrogram.api.types.TopPeerCategoryChannels* attribute), 427
- ID (*pyrogram.api.types.TopPeerCategoryCorrespondents* attribute), 427
- ID (*pyrogram.api.types.TopPeerCategoryGroups* attribute), 427
- ID (*pyrogram.api.types.TopPeerCategoryPeers* attribute), 428
- ID (*pyrogram.api.types.TopPeerCategoryPhoneCalls* attribute), 428
- ID (*pyrogram.api.types.Update* attribute), 428
- ID (*pyrogram.api.types.UpdateBotCallbackQuery* attribute), 429
- ID (*pyrogram.api.types.UpdateBotInlineQuery* attribute), 430
- ID (*pyrogram.api.types.UpdateBotInlineSend* attribute), 430
- ID (*pyrogram.api.types.UpdateBotPrecheckoutQuery* attribute), 430
- ID (*pyrogram.api.types.UpdateBotShippingQuery* attribute), 431
- ID (*pyrogram.api.types.UpdateBotWebhookJSON* attribute), 431
- ID (*pyrogram.api.types.UpdateBotWebhookJSONQuery* attribute), 431
- ID (*pyrogram.api.types.UpdateChannel* attribute), 431
- ID (*pyrogram.api.types.UpdateChannelAvailableMessages* attribute), 432
- ID (*pyrogram.api.types.UpdateChannelMessageViews* attribute), 432
- ID (*pyrogram.api.types.UpdateChannelPinnedMessage* attribute), 432
- ID (*pyrogram.api.types.UpdateChannelReadMessagesContents* attribute), 432
- ID (*pyrogram.api.types.UpdateChannelTooLong* attribute), 433
- ID (*pyrogram.api.types.UpdateChannelWebPage* attribute), 433
- ID (*pyrogram.api.types.UpdateChatDefaultBannedRights* attribute), 433
- ID (*pyrogram.api.types.UpdateChatParticipantAdd* attribute), 434
- ID (*pyrogram.api.types.UpdateChatParticipantAdmin* attribute), 434
- ID (*pyrogram.api.types.UpdateChatParticipantDelete* attribute), 434
- ID (*pyrogram.api.types.UpdateChatParticipants* attribute), 435
- ID (*pyrogram.api.types.UpdateChatPinnedMessage* attribute), 435
- ID (*pyrogram.api.types.UpdateChatUserTyping* attribute), 435
- ID (*pyrogram.api.types.UpdateConfig* attribute), 435
- ID (*pyrogram.api.types.UpdateContactLink* attribute), 436
- ID (*pyrogram.api.types.UpdateContactsReset* attribute), 436
- ID (*pyrogram.api.types.UpdateDcOptions* attribute), 436
- ID (*pyrogram.api.types.UpdateDeleteChannelMessages* attribute), 436
- ID (*pyrogram.api.types.UpdateDeleteMessages* attribute), 437
- ID (*pyrogram.api.types.UpdateDialogPinned* attribute), 437
- ID (*pyrogram.api.types.UpdateDialogUnreadMark* attribute), 437
- ID (*pyrogram.api.types.UpdateDraftMessage* attribute), 437
- ID (*pyrogram.api.types.UpdateEditChannelMessage* attribute), 438
- ID (*pyrogram.api.types.UpdateEditMessage* attribute), 438
- ID (*pyrogram.api.types.UpdateEncryptedChatTyping* attribute), 438
- ID (*pyrogram.api.types.UpdateEncryptedMessagesRead* attribute), 438
- ID (*pyrogram.api.types.UpdateEncryption* attribute), 439
- ID (*pyrogram.api.types.UpdateFavedStickers* attribute), 439
- ID (*pyrogram.api.types.UpdateInlineBotCallbackQuery* attribute), 439

- ID (*pyrogram.api.types.UpdateLangPack* attribute), 439
- ID (*pyrogram.api.types.UpdateLangPackTooLong* attribute), 440
- ID (*pyrogram.api.types.UpdateMessageID* attribute), 440
- ID (*pyrogram.api.types.UpdateMessagePoll* attribute), 440
- ID (*pyrogram.api.types.UpdateNewChannelMessage* attribute), 440
- ID (*pyrogram.api.types.UpdateNewEncryptedMessage* attribute), 441
- ID (*pyrogram.api.types.UpdateNewMessage* attribute), 441
- ID (*pyrogram.api.types.UpdateNewStickerSet* attribute), 441
- ID (*pyrogram.api.types.UpdateNotifySettings* attribute), 441
- ID (*pyrogram.api.types.UpdatePhoneCall* attribute), 442
- ID (*pyrogram.api.types.UpdatePinnedDialogs* attribute), 442
- ID (*pyrogram.api.types.UpdatePrivacy* attribute), 442
- ID (*pyrogram.api.types.UpdatePtsChanged* attribute), 442
- ID (*pyrogram.api.types.UpdateReadChannelInbox* attribute), 443
- ID (*pyrogram.api.types.UpdateReadChannelOutbox* attribute), 443
- ID (*pyrogram.api.types.UpdateReadFeaturedStickers* attribute), 443
- ID (*pyrogram.api.types.UpdateReadHistoryInbox* attribute), 443
- ID (*pyrogram.api.types.UpdateReadHistoryOutbox* attribute), 444
- ID (*pyrogram.api.types.UpdateReadMessagesContents* attribute), 444
- ID (*pyrogram.api.types.UpdateRecentStickers* attribute), 444
- ID (*pyrogram.api.types.updates.ChannelDifference* attribute), 270
- ID (*pyrogram.api.types.updates.ChannelDifferenceEmpty* attribute), 271
- ID (*pyrogram.api.types.updates.ChannelDifferenceTooLong* attribute), 271
- ID (*pyrogram.api.types.updates.Difference* attribute), 272
- ID (*pyrogram.api.types.updates.DifferenceEmpty* attribute), 273
- ID (*pyrogram.api.types.updates.DifferenceSlice* attribute), 273
- ID (*pyrogram.api.types.updates.DifferenceTooLong* attribute), 274
- ID (*pyrogram.api.types.updates.State* attribute), 274
- ID (*pyrogram.api.types.UpdateSavedGifs* attribute), 444
- ID (*pyrogram.api.types.UpdatesCombined* attribute), 452
- ID (*pyrogram.api.types.UpdateServiceNotification* attribute), 445
- ID (*pyrogram.api.types.UpdateShort* attribute), 445
- ID (*pyrogram.api.types.UpdateShortChatMessage* attribute), 446
- ID (*pyrogram.api.types.UpdateShortMessage* attribute), 447
- ID (*pyrogram.api.types.UpdateShortSentMessage* attribute), 448
- ID (*pyrogram.api.types.UpdateStickerSets* attribute), 449
- ID (*pyrogram.api.types.UpdateStickerSetsOrder* attribute), 449
- ID (*pyrogram.api.types.UpdatesTooLong* attribute), 453
- ID (*pyrogram.api.types.UpdateUserBlocked* attribute), 449
- ID (*pyrogram.api.types.UpdateUserName* attribute), 450
- ID (*pyrogram.api.types.UpdateUserPhone* attribute), 450
- ID (*pyrogram.api.types.UpdateUserPhoto* attribute), 450
- ID (*pyrogram.api.types.UpdateUserPinnedMessage* attribute), 451
- ID (*pyrogram.api.types.UpdateUserStatus* attribute), 451
- ID (*pyrogram.api.types.UpdateUserTyping* attribute), 451
- ID (*pyrogram.api.types.UpdateWebPage* attribute), 451
- ID (*pyrogram.api.types.upload.CdnFile* attribute), 275
- ID (*pyrogram.api.types.upload.CdnFileReuploadNeeded* attribute), 275
- ID (*pyrogram.api.types.upload.File* attribute), 275
- ID (*pyrogram.api.types.upload.FileCdnRedirect* attribute), 276
- ID (*pyrogram.api.types.upload.WebFile* attribute), 276
- ID (*pyrogram.api.types.User* attribute), 453
- ID (*pyrogram.api.types.UserEmpty* attribute), 454
- ID (*pyrogram.api.types.UserFull* attribute), 455
- ID (*pyrogram.api.types.UserProfilePhoto* attribute), 455
- ID (*pyrogram.api.types.UserProfilePhotoEmpty* attribute), 456
- ID (*pyrogram.api.types.UserStatusEmpty* attribute), 456
- ID (*pyrogram.api.types.UserStatusLastMonth* attribute), 456
- ID (*pyrogram.api.types.UserStatusLastWeek* attribute), 456
- ID (*pyrogram.api.types.UserStatusOffline* attribute), 456
- ID (*pyrogram.api.types.UserStatusOnline* attribute), 457
- ID (*pyrogram.api.types.UserStatusRecently* attribute), 457
- ID (*pyrogram.api.types.WallPaper* attribute), 457
- ID (*pyrogram.api.types.WallPaperSettings* attribute), 457
- ID (*pyrogram.api.types.WebAuthorization* attribute), 458
- ID (*pyrogram.api.types.WebDocument* attribute), 458
- ID (*pyrogram.api.types.WebDocumentNoProxy* attribute), 459

- ID (*pyrogram.api.types.WebPage* attribute), 459
- ID (*pyrogram.api.types.WebPageEmpty* attribute), 460
- ID (*pyrogram.api.types.WebPageNotModified* attribute), 460
- ID (*pyrogram.api.types.WebPagePending* attribute), 460
- idle() (*pyrogram.Client* method), 23
- ImportAuthorization (class in *pyrogram.api.functions.auth*), 159
- ImportBotAuthorization (class in *pyrogram.api.functions.auth*), 159
- ImportChatInvite (class in *pyrogram.api.functions.messages*), 200
- ImportContacts (class in *pyrogram.api.functions.contacts*), 174
- ImportedContact (class in *pyrogram.api.types*), 318
- ImportedContacts (class in *pyrogram.api.types.contacts*), 243
- incoming (*pyrogram.Filters* attribute), 103
- InitConnection (class in *pyrogram.api.functions*), 228
- InitTakeoutSession (class in *pyrogram.api.functions.account*), 150
- inline\_keyboard (*pyrogram.Filters* attribute), 105
- InlineBotSwitchPM (class in *pyrogram.api.types*), 318
- InlineKeyboardButton (class in *pyrogram*), 69
- InlineKeyboardMarkup (class in *pyrogram*), 69
- InlineQuery (class in *pyrogram*), 73
- InlineQueryHandler (class in *pyrogram*), 99
- InlineQueryResult (class in *pyrogram*), 73
- InlineQueryResultArticle (class in *pyrogram*), 73
- InputAppEvent (class in *pyrogram.api.types*), 318
- InputBotInlineMessageGame (class in *pyrogram.api.types*), 318
- InputBotInlineMessageID (class in *pyrogram.api.types*), 319
- InputBotInlineMessageMediaAuto (class in *pyrogram.api.types*), 319
- InputBotInlineMessageMediaContact (class in *pyrogram.api.types*), 319
- InputBotInlineMessageMediaGeo (class in *pyrogram.api.types*), 320
- InputBotInlineMessageMediaVenue (class in *pyrogram.api.types*), 320
- InputBotInlineMessageText (class in *pyrogram.api.types*), 321
- InputBotInlineResult (class in *pyrogram.api.types*), 321
- InputBotInlineResultDocument (class in *pyrogram.api.types*), 321
- InputBotInlineResultGame (class in *pyrogram.api.types*), 322
- InputBotInlineResultPhoto (class in *pyrogram.api.types*), 322
- InputChannel (class in *pyrogram.api.types*), 323
- InputChannelEmpty (class in *pyrogram.api.types*), 323
- InputChatPhoto (class in *pyrogram.api.types*), 323
- InputChatPhotoEmpty (class in *pyrogram.api.types*), 323
- InputChatUploadedPhoto (class in *pyrogram.api.types*), 323
- InputCheckPasswordEmpty (class in *pyrogram.api.types*), 324
- InputCheckPasswordSRP (class in *pyrogram.api.types*), 324
- InputClientProxy (class in *pyrogram.api.types*), 324
- InputDialogPeer (class in *pyrogram.api.types*), 324
- InputDocument (class in *pyrogram.api.types*), 325
- InputDocumentEmpty (class in *pyrogram.api.types*), 325
- InputDocumentFileLocation (class in *pyrogram.api.types*), 325
- InputEncryptedChat (class in *pyrogram.api.types*), 325
- InputEncryptedFile (class in *pyrogram.api.types*), 326
- InputEncryptedFileBigUploaded (class in *pyrogram.api.types*), 326
- InputEncryptedFileEmpty (class in *pyrogram.api.types*), 326
- InputEncryptedFileLocation (class in *pyrogram.api.types*), 326
- InputEncryptedFileUploaded (class in *pyrogram.api.types*), 327
- InputFile (class in *pyrogram.api.types*), 327
- InputFileBig (class in *pyrogram.api.types*), 327
- InputFileLocation (class in *pyrogram.api.types*), 327
- InputGameID (class in *pyrogram.api.types*), 328
- InputGameShortName (class in *pyrogram.api.types*), 328
- InputGeoPoint (class in *pyrogram.api.types*), 328
- InputGeoPointEmpty (class in *pyrogram.api.types*), 329
- InputMedia (class in *pyrogram*), 71
- InputMediaAnimation (class in *pyrogram*), 72
- InputMediaAudio (class in *pyrogram*), 71
- InputMediaContact (class in *pyrogram.api.types*), 329
- InputMediaDocument (class in *pyrogram*), 72
- InputMediaDocument (class in *pyrogram.api.types*), 329
- InputMediaDocumentExternal (class in *pyrogram.api.types*), 329
- InputMediaEmpty (class in *pyrogram.api.types*), 330



- [InputMediaGame \(class in pyrogram.api.types\), 330](#)  
[InputMediaGeoLive \(class in pyrogram.api.types\), 330](#)  
[InputMediaGeoPoint \(class in pyrogram.api.types\), 330](#)  
[InputMediaGifExternal \(class in pyrogram.api.types\), 330](#)  
[InputMediaInvoice \(class in pyrogram.api.types\), 331](#)  
[InputMediaPhoto \(class in pyrogram\), 71](#)  
[InputMediaPhoto \(class in pyrogram.api.types\), 331](#)  
[InputMediaPhotoExternal \(class in pyrogram.api.types\), 331](#)  
[InputMediaPoll \(class in pyrogram.api.types\), 332](#)  
[InputMediaUploadedDocument \(class in pyrogram.api.types\), 332](#)  
[InputMediaUploadedPhoto \(class in pyrogram.api.types\), 332](#)  
[InputMediaVenue \(class in pyrogram.api.types\), 332](#)  
[InputMediaVideo \(class in pyrogram\), 71](#)  
[InputMessageContent \(class in pyrogram\), 73](#)  
[InputMessageEntityMentionName \(class in pyrogram.api.types\), 333](#)  
[InputMessageID \(class in pyrogram.api.types\), 333](#)  
[InputMessagePinned \(class in pyrogram.api.types\), 333](#)  
[InputMessageReplyTo \(class in pyrogram.api.types\), 334](#)  
[InputMessagesFilterChatPhotos \(class in pyrogram.api.types\), 334](#)  
[InputMessagesFilterContacts \(class in pyrogram.api.types\), 334](#)  
[InputMessagesFilterDocument \(class in pyrogram.api.types\), 334](#)  
[InputMessagesFilterEmpty \(class in pyrogram.api.types\), 334](#)  
[InputMessagesFilterGeo \(class in pyrogram.api.types\), 335](#)  
[InputMessagesFilterGif \(class in pyrogram.api.types\), 335](#)  
[InputMessagesFilterMusic \(class in pyrogram.api.types\), 335](#)  
[InputMessagesFilterMyMentions \(class in pyrogram.api.types\), 335](#)  
[InputMessagesFilterPhoneCalls \(class in pyrogram.api.types\), 335](#)  
[InputMessagesFilterPhotos \(class in pyrogram.api.types\), 336](#)  
[InputMessagesFilterPhotoVideo \(class in pyrogram.api.types\), 336](#)  
[InputMessagesFilterRoundVideo \(class in pyrogram.api.types\), 336](#)  
[InputMessagesFilterRoundVoice \(class in pyrogram.api.types\), 336](#)  
[InputMessagesFilterUrl \(class in pyrogram.api.types\), 336](#)  
[InputMessagesFilterVideo \(class in pyrogram.api.types\), 337](#)  
[InputMessagesFilterVoice \(class in pyrogram.api.types\), 337](#)  
[InputNotifyBroadcasts \(class in pyrogram.api.types\), 337](#)  
[InputNotifyChats \(class in pyrogram.api.types\), 337](#)  
[InputNotifyPeer \(class in pyrogram.api.types\), 337](#)  
[InputNotifyUsers \(class in pyrogram.api.types\), 338](#)  
[InputPaymentCredentials \(class in pyrogram.api.types\), 338](#)  
[InputPaymentCredentialsAndroidPay \(class in pyrogram.api.types\), 338](#)  
[InputPaymentCredentialsApplePay \(class in pyrogram.api.types\), 338](#)  
[InputPaymentCredentialsSaved \(class in pyrogram.api.types\), 338](#)  
[InputPeerChannel \(class in pyrogram.api.types\), 339](#)  
[InputPeerChat \(class in pyrogram.api.types\), 339](#)  
[InputPeerEmpty \(class in pyrogram.api.types\), 339](#)  
[InputPeerNotifySettings \(class in pyrogram.api.types\), 339](#)  
[InputPeerSelf \(class in pyrogram.api.types\), 340](#)  
[InputPeerUser \(class in pyrogram.api.types\), 340](#)  
[InputPhoneCall \(class in pyrogram.api.types\), 340](#)  
[InputPhoneContact \(class in pyrogram\), 72](#)  
[InputPhoneContact \(class in pyrogram.api.types\), 340](#)  
[InputPhoto \(class in pyrogram.api.types\), 341](#)  
[InputPhotoEmpty \(class in pyrogram.api.types\), 341](#)  
[InputPrivacyKeyChatInvite \(class in pyrogram.api.types\), 341](#)  
[InputPrivacyKeyForwards \(class in pyrogram.api.types\), 341](#)  
[InputPrivacyKeyPhoneCall \(class in pyrogram.api.types\), 341](#)  
[InputPrivacyKeyPhoneP2P \(class in pyrogram.api.types\), 342](#)  
[InputPrivacyKeyProfilePhoto \(class in pyrogram.api.types\), 342](#)  
[InputPrivacyKeyStatusTimestamp \(class in pyrogram.api.types\), 342](#)  
[InputPrivacyValueAllowAll \(class in pyrogram.api.types\), 342](#)  
[InputPrivacyValueAllowContacts \(class in pyrogram.api.types\), 342](#)  
[InputPrivacyValueAllowUsers \(class in pyrogram.api.types\), 343](#)  
[InputPrivacyValueDisallowAll \(class in pyrogram\)](#)

- gram.api.types*), 343
- InputPrivacyValueDisallowContacts* (class in *pyrogram.api.types*), 343
- InputPrivacyValueDisallowUsers* (class in *pyrogram.api.types*), 343
- InputReportReasonChildAbuse* (class in *pyrogram.api.types*), 343
- InputReportReasonCopyright* (class in *pyrogram.api.types*), 344
- InputReportReasonOther* (class in *pyrogram.api.types*), 344
- InputReportReasonPornography* (class in *pyrogram.api.types*), 344
- InputReportReasonSpam* (class in *pyrogram.api.types*), 344
- InputReportReasonViolence* (class in *pyrogram.api.types*), 344
- InputSecureFile* (class in *pyrogram.api.types*), 345
- InputSecureFileLocation* (class in *pyrogram.api.types*), 345
- InputSecureFileUploaded* (class in *pyrogram.api.types*), 345
- InputSecureValue* (class in *pyrogram.api.types*), 345
- InputSingleMedia* (class in *pyrogram.api.types*), 346
- InputStickeredMediaDocument* (class in *pyrogram.api.types*), 347
- InputStickeredMediaPhoto* (class in *pyrogram.api.types*), 348
- InputStickerSetEmpty* (class in *pyrogram.api.types*), 347
- InputStickerSetID* (class in *pyrogram.api.types*), 347
- InputStickerSetItem* (class in *pyrogram.api.types*), 347
- InputStickerSetShortName* (class in *pyrogram.api.types*), 347
- InputTakeoutFileLocation* (class in *pyrogram.api.types*), 348
- InputTextMessageContent* (class in *pyrogram*), 73
- InputUser* (class in *pyrogram.api.types*), 348
- InputUserEmpty* (class in *pyrogram.api.types*), 348
- InputUserSelf* (class in *pyrogram.api.types*), 349
- InputWallPaper* (class in *pyrogram.api.types*), 349
- InputWallPaperSlug* (class in *pyrogram.api.types*), 349
- InputWebDocument* (class in *pyrogram.api.types*), 349
- InputWebFileGeoPointLocation* (class in *pyrogram.api.types*), 350
- InputWebFileLocation* (class in *pyrogram.api.types*), 350
- InstallStickerSet* (class in *pyrogram.api.functions.messages*), 201
- InstallWallPaper* (class in *pyrogram.api.functions.account*), 150
- InviteText* (class in *pyrogram.api.types.help*), 247
- InviteToChannel* (class in *pyrogram.api.functions.channels*), 168
- Invoice* (class in *pyrogram.api.types*), 350
- InvokeAfterMsg* (class in *pyrogram.api.functions*), 229
- InvokeAfterMsgs* (class in *pyrogram.api.functions*), 229
- InvokeWithLayer* (class in *pyrogram.api.functions*), 229
- InvokeWithMessagesRange* (class in *pyrogram.api.functions*), 230
- InvokeWithoutUpdates* (class in *pyrogram.api.functions*), 230
- InvokeWithTakeout* (class in *pyrogram.api.functions*), 230
- IpPort* (class in *pyrogram.api.types*), 351
- IpPortSecret* (class in *pyrogram.api.types*), 351
- iter\_chat\_members()* (*pyrogram.Client* method), 47
- iter\_dialogs()* (*pyrogram.Client* method), 48
- iter\_history()* (*pyrogram.Client* method), 39
- ## J
- join\_chat()* (*pyrogram.Client* method), 41
- JoinChannel* (class in *pyrogram.api.functions.channels*), 169
- JSONArray* (class in *pyrogram.api.types*), 351
- JsonBool* (class in *pyrogram.api.types*), 351
- JsonNull* (class in *pyrogram.api.types*), 352
- JsonNumber* (class in *pyrogram.api.types*), 352
- JsonObject* (class in *pyrogram.api.types*), 352
- JsonObjectValue* (class in *pyrogram.api.types*), 352
- JsonString* (class in *pyrogram.api.types*), 353
- ## K
- KeyboardButton* (class in *pyrogram*), 69
- KeyboardButton* (class in *pyrogram.api.types*), 353
- KeyboardButtonBuy* (class in *pyrogram.api.types*), 353
- KeyboardButtonCallback* (class in *pyrogram.api.types*), 353
- KeyboardButtonGame* (class in *pyrogram.api.types*), 354
- KeyboardButtonRequestGeoLocation* (class in *pyrogram.api.types*), 354
- KeyboardButtonRequestPhone* (class in *pyrogram.api.types*), 354
- KeyboardButtonRow* (class in *pyrogram.api.types*), 354

KeyboardButtonSwitchInline (class in *pyrogram.api.types*), 354  
 KeyboardButtonUrl (class in *pyrogram.api.types*), 355  
 kick\_chat\_member() (*pyrogram.Client* method), 42

## L

LabeledPrice (class in *pyrogram.api.types*), 355  
 LangPackDifference (class in *pyrogram.api.types*), 355  
 LangPackLanguage (class in *pyrogram.api.types*), 356  
 LangPackString (class in *pyrogram.api.types*), 356  
 LangPackStringDeleted (class in *pyrogram.api.types*), 356  
 LangPackStringPluralized (class in *pyrogram.api.types*), 357  
 leave\_chat() (*pyrogram.Client* method), 42  
 LeaveChannel (class in *pyrogram.api.functions.channels*), 169  
 left\_chat\_member (*pyrogram.Filters* attribute), 105  
 Link (class in *pyrogram.api.types.contacts*), 243  
 Location (class in *pyrogram*), 67  
 location (*pyrogram.Filters* attribute), 104  
 LogOut (class in *pyrogram.api.functions.auth*), 160

## M

MarkDialogUnread (class in *pyrogram.api.functions.messages*), 201  
 MaskCoords (class in *pyrogram.api.types*), 357  
 me (*pyrogram.Filters* attribute), 103  
 media (*pyrogram.Filters* attribute), 106  
 media\_group (*pyrogram.Filters* attribute), 104  
 mentioned (*pyrogram.Filters* attribute), 105  
 Message (class in *pyrogram*), 61  
 Message (class in *pyrogram.api.types*), 357  
 MessageActionBotAllowed (class in *pyrogram.api.types*), 358  
 MessageActionChannelCreate (class in *pyrogram.api.types*), 359  
 MessageActionChannelMigrateFrom (class in *pyrogram.api.types*), 359  
 MessageActionChatAddUser (class in *pyrogram.api.types*), 359  
 MessageActionChatCreate (class in *pyrogram.api.types*), 359  
 MessageActionChatDeletePhoto (class in *pyrogram.api.types*), 360  
 MessageActionChatDeleteUser (class in *pyrogram.api.types*), 360  
 MessageActionChatEditPhoto (class in *pyrogram.api.types*), 360  
 MessageActionChatEditTitle (class in *pyrogram.api.types*), 360

MessageActionChatJoinedByLink (class in *pyrogram.api.types*), 360  
 MessageActionChatMigrateTo (class in *pyrogram.api.types*), 361  
 MessageActionContactSignUp (class in *pyrogram.api.types*), 361  
 MessageActionCustomAction (class in *pyrogram.api.types*), 361  
 MessageActionEmpty (class in *pyrogram.api.types*), 361  
 MessageActionGameScore (class in *pyrogram.api.types*), 361  
 MessageActionHistoryClear (class in *pyrogram.api.types*), 362  
 MessageActionPaymentSent (class in *pyrogram.api.types*), 362  
 MessageActionPaymentSentMe (class in *pyrogram.api.types*), 362  
 MessageActionPhoneCall (class in *pyrogram.api.types*), 362  
 MessageActionPinMessage (class in *pyrogram.api.types*), 363  
 MessageActionScreenshotTaken (class in *pyrogram.api.types*), 363  
 MessageActionSecureValuesSent (class in *pyrogram.api.types*), 363  
 MessageActionSecureValuesSentMe (class in *pyrogram.api.types*), 363  
 MessageEditData (class in *pyrogram.api.types.messages*), 259  
 MessageEmpty (class in *pyrogram.api.types*), 364  
 MessageEntity (class in *pyrogram*), 64  
 MessageEntityBold (class in *pyrogram.api.types*), 364  
 MessageEntityBotCommand (class in *pyrogram.api.types*), 364  
 MessageEntityCashtag (class in *pyrogram.api.types*), 364  
 MessageEntityCode (class in *pyrogram.api.types*), 365  
 MessageEntityEmail (class in *pyrogram.api.types*), 365  
 MessageEntityHashtag (class in *pyrogram.api.types*), 365  
 MessageEntityItalic (class in *pyrogram.api.types*), 365  
 MessageEntityMention (class in *pyrogram.api.types*), 366  
 MessageEntityMentionName (class in *pyrogram.api.types*), 366  
 MessageEntityPhone (class in *pyrogram.api.types*), 366  
 MessageEntityPre (class in *pyrogram.api.types*), 366

- MessageEntityTextUrl (class in *pyrogram.api.types*), 367  
 MessageEntityUnknown (class in *pyrogram.api.types*), 367  
 MessageEntityUrl (class in *pyrogram.api.types*), 367  
 MessageFwdHeader (class in *pyrogram.api.types*), 367  
 MessageHandler (class in *pyrogram*), 98  
 MessageMediaContact (class in *pyrogram.api.types*), 368  
 MessageMediaDocument (class in *pyrogram.api.types*), 368  
 MessageMediaEmpty (class in *pyrogram.api.types*), 369  
 MessageMediaGame (class in *pyrogram.api.types*), 369  
 MessageMediaGeo (class in *pyrogram.api.types*), 369  
 MessageMediaGeoLive (class in *pyrogram.api.types*), 369  
 MessageMediaInvoice (class in *pyrogram.api.types*), 370  
 MessageMediaPhoto (class in *pyrogram.api.types*), 370  
 MessageMediaPoll (class in *pyrogram.api.types*), 371  
 MessageMediaUnsupported (class in *pyrogram.api.types*), 371  
 MessageMediaVenue (class in *pyrogram.api.types*), 371  
 MessageMediaWebPage (class in *pyrogram.api.types*), 372  
 MessageRange (class in *pyrogram.api.types*), 372  
 Messages (class in *pyrogram*), 64  
 Messages (class in *pyrogram.api.types.messages*), 259  
 MessageService (class in *pyrogram.api.types*), 372  
 MessagesNotModified (class in *pyrogram.api.types.messages*), 260  
 MessagesSlice (class in *pyrogram.api.types.messages*), 260  
 migrate\_from\_chat\_id (*pyrogram.Filters* attribute), 105  
 migrate\_to\_chat\_id (*pyrogram.Filters* attribute), 105  
 MigrateChat (class in *pyrogram.api.functions.messages*), 201  
 MsgDetailedInfo (class in *pyrogram.api.types*), 373  
 MsgNewDetailedInfo (class in *pyrogram.api.types*), 373  
 MsgResendAnsReq (class in *pyrogram.api.types*), 373  
 MsgResendReq (class in *pyrogram.api.types*), 374  
 MsgsAck (class in *pyrogram.api.types*), 374  
 MsgsAllInfo (class in *pyrogram.api.types*), 374  
 MsgsStateInfo (class in *pyrogram.api.types*), 374  
 MsgsStateReq (class in *pyrogram.api.types*), 375  
 MTPProto, 141  
 MTPProto API, 141
- ## N
- NearestDc (class in *pyrogram.api.types*), 375  
 new\_chat\_members (*pyrogram.Filters* attribute), 105  
 new\_chat\_photo (*pyrogram.Filters* attribute), 105  
 new\_chat\_title (*pyrogram.Filters* attribute), 105  
 NewSessionCreated (class in *pyrogram.api.types*), 375  
 NoAppUpdate (class in *pyrogram.api.types.help*), 247  
 NotifyBroadcasts (class in *pyrogram.api.types*), 375  
 NotifyChats (class in *pyrogram.api.types*), 376  
 NotifyPeer (class in *pyrogram.api.types*), 376  
 NotifyUsers (class in *pyrogram.api.types*), 376
- ## O
- on\_callback\_query() (*pyrogram.Client* method), 102  
 on\_deleted\_messages() (*pyrogram.Client* method), 102  
 on\_disconnect() (*pyrogram.Client* method), 103  
 on\_inline\_query() (*pyrogram.Client* method), 102  
 on\_message() (*pyrogram.Client* method), 102  
 on\_poll() (*pyrogram.Client* method), 102  
 on\_raw\_update() (*pyrogram.Client* method), 103  
 on\_user\_status() (*pyrogram.Client* method), 102  
 outgoing (*pyrogram.Filters* attribute), 104
- ## P
- Page (class in *pyrogram.api.types*), 378  
 PageBlockAnchor (class in *pyrogram.api.types*), 378  
 PageBlockAudio (class in *pyrogram.api.types*), 379  
 PageBlockAuthorDate (class in *pyrogram.api.types*), 379  
 PageBlockBlockquote (class in *pyrogram.api.types*), 379  
 PageBlockChannel (class in *pyrogram.api.types*), 379  
 PageBlockCollage (class in *pyrogram.api.types*), 380  
 PageBlockCover (class in *pyrogram.api.types*), 380  
 PageBlockDetails (class in *pyrogram.api.types*), 380  
 PageBlockDivider (class in *pyrogram.api.types*), 381  
 PageBlockEmbed (class in *pyrogram.api.types*), 381  
 PageBlockEmbedPost (class in *pyrogram.api.types*), 382  
 PageBlockFooter (class in *pyrogram.api.types*), 382  
 PageBlockHeader (class in *pyrogram.api.types*), 382  
 PageBlockKicker (class in *pyrogram.api.types*), 383



- PageBlockList (class in *pyrogram.api.types*), 383  
 PageBlockMap (class in *pyrogram.api.types*), 383  
 PageBlockOrderedList (class in *pyrogram.api.types*), 383  
 PageBlockParagraph (class in *pyrogram.api.types*), 384  
 PageBlockPhoto (class in *pyrogram.api.types*), 384  
 PageBlockPreformatted (class in *pyrogram.api.types*), 384  
 PageBlockPullquote (class in *pyrogram.api.types*), 385  
 PageBlockRelatedArticles (class in *pyrogram.api.types*), 385  
 PageBlockSlideshow (class in *pyrogram.api.types*), 385  
 PageBlockSubheader (class in *pyrogram.api.types*), 386  
 PageBlockSubtitle (class in *pyrogram.api.types*), 386  
 PageBlockTable (class in *pyrogram.api.types*), 386  
 PageBlockTitle (class in *pyrogram.api.types*), 387  
 PageBlockUnsupported (class in *pyrogram.api.types*), 387  
 PageBlockVideo (class in *pyrogram.api.types*), 387  
 PageCaption (class in *pyrogram.api.types*), 387  
 PageListItemBlocks (class in *pyrogram.api.types*), 388  
 PageListItemText (class in *pyrogram.api.types*), 388  
 PageListOrderedItemBlocks (class in *pyrogram.api.types*), 388  
 PageListOrderedItemText (class in *pyrogram.api.types*), 389  
 PageRelatedArticle (class in *pyrogram.api.types*), 389  
 PageTableCell (class in *pyrogram.api.types*), 389  
 PageTableRow (class in *pyrogram.api.types*), 390  
 PassportConfig (class in *pyrogram.api.types.help*), 247  
 PassportConfigNotModified (class in *pyrogram.api.types.help*), 247  
 Password (class in *pyrogram.api.types.account*), 234  
 PasswordInputSettings (class in *pyrogram.api.types.account*), 234  
 PasswordKdfAlgoSHA256SHA256PBKDF2HMACSHA256 (class in *pyrogram.api.types*), 390  
 PasswordKdfAlgoUnknown (class in *pyrogram.api.types*), 390  
 PasswordRecovery (class in *pyrogram.api.types.auth*), 238  
 PasswordSettings (class in *pyrogram.api.types.account*), 235  
 PaymentCharge (class in *pyrogram.api.types*), 391  
 PaymentForm (class in *pyrogram.api.types.payments*), 264  
 PaymentReceipt (class in *pyrogram.api.types.payments*), 265  
 PaymentRequestedInfo (class in *pyrogram.api.types*), 391  
 PaymentResult (class in *pyrogram.api.types.payments*), 265  
 PaymentSavedCredentialsCard (class in *pyrogram.api.types*), 391  
 PaymentVerificationNeeded (class in *pyrogram.api.types.payments*), 265  
 PeerChannel (class in *pyrogram.api.types*), 391  
 PeerChat (class in *pyrogram.api.types*), 392  
 PeerDialogs (class in *pyrogram.api.types.messages*), 260  
 PeerNotifySettings (class in *pyrogram.api.types*), 392  
 PeerSettings (class in *pyrogram.api.types*), 392  
 PeerUser (class in *pyrogram.api.types*), 392  
 PhoneCall (class in *pyrogram.api.types*), 393  
 PhoneCall (class in *pyrogram.api.types.phone*), 266  
 PhoneCallAccepted (class in *pyrogram.api.types*), 393  
 PhoneCallDiscarded (class in *pyrogram.api.types*), 394  
 PhoneCallDiscardReasonBusy (class in *pyrogram.api.types*), 394  
 PhoneCallDiscardReasonDisconnect (class in *pyrogram.api.types*), 394  
 PhoneCallDiscardReasonHangup (class in *pyrogram.api.types*), 394  
 PhoneCallDiscardReasonMissed (class in *pyrogram.api.types*), 394  
 PhoneCallEmpty (class in *pyrogram.api.types*), 395  
 PhoneCallProtocol (class in *pyrogram.api.types*), 395  
 PhoneCallRequested (class in *pyrogram.api.types*), 395  
 PhoneCallWaiting (class in *pyrogram.api.types*), 396  
 PhoneConnection (class in *pyrogram.api.types*), 396  
 Photo (class in *pyrogram*), 64  
 Photo (class in *pyrogram.api.types*), 396  
 Photo (class in *pyrogram.api.types.photos*), 267  
 PhotoCachedSize (class in *pyrogram.api.types*), 397  
 PhotoEmpty (class in *pyrogram.api.types*), 397  
 Photos (class in *pyrogram.api.types.photos*), 267  
 PhotoSize (class in *pyrogram*), 64  
 PhotoSize (class in *pyrogram.api.types*), 397  
 PhotoSizeEmpty (class in *pyrogram.api.types*), 398  
 PhotosSlice (class in *pyrogram.api.types.photos*), 267  
 PhotoStrippedSize (class in *pyrogram.api.types*), 267

- 398  
[pin\(\)](#) (*pyrogram.Message* method), 80  
[pin\\_chat\\_message\(\)](#) (*pyrogram.Client* method), 45  
[Ping](#) (*class in pyrogram.api.functions*), 231  
[PingDelayDisconnect](#) (*class in pyrogram.api.functions*), 231  
[pinned\\_message](#) (*pyrogram.Filters* attribute), 105  
[Poll](#) (*class in pyrogram*), 68  
[Poll](#) (*class in pyrogram.api.types*), 398  
[poll](#) (*pyrogram.Filters* attribute), 104  
[PollAnswer](#) (*class in pyrogram.api.types*), 398  
[PollAnswerVoters](#) (*class in pyrogram.api.types*), 399  
[PollHandler](#) (*class in pyrogram*), 100  
[PollOption](#) (*class in pyrogram*), 68  
[PollResults](#) (*class in pyrogram.api.types*), 399  
[Pong](#) (*class in pyrogram.api.types*), 399  
[PopularContact](#) (*class in pyrogram.api.types*), 399  
[PostAddress](#) (*class in pyrogram.api.types*), 400  
[PQInnerData](#) (*class in pyrogram.api.types*), 376  
[PQInnerDataDc](#) (*class in pyrogram.api.types*), 377  
[PQInnerDataTemp](#) (*class in pyrogram.api.types*), 377  
[PQInnerDataTempDc](#) (*class in pyrogram.api.types*), 377  
[PrivacyKeyChatInvite](#) (*class in pyrogram.api.types*), 400  
[PrivacyKeyForwards](#) (*class in pyrogram.api.types*), 400  
[PrivacyKeyPhoneCall](#) (*class in pyrogram.api.types*), 400  
[PrivacyKeyPhoneP2P](#) (*class in pyrogram.api.types*), 401  
[PrivacyKeyProfilePhoto](#) (*class in pyrogram.api.types*), 401  
[PrivacyKeyStatusTimestamp](#) (*class in pyrogram.api.types*), 401  
[PrivacyRules](#) (*class in pyrogram.api.types.account*), 235  
[PrivacyValueAllowAll](#) (*class in pyrogram.api.types*), 401  
[PrivacyValueAllowContacts](#) (*class in pyrogram.api.types*), 401  
[PrivacyValueAllowUsers](#) (*class in pyrogram.api.types*), 402  
[PrivacyValueDisallowAll](#) (*class in pyrogram.api.types*), 402  
[PrivacyValueDisallowContacts](#) (*class in pyrogram.api.types*), 402  
[PrivacyValueDisallowUsers](#) (*class in pyrogram.api.types*), 402  
[private](#) (*pyrogram.Filters* attribute), 105  
[promote\\_chat\\_member\(\)](#) (*pyrogram.Client* method), 43  
[ProxyDataEmpty](#) (*class in pyrogram.api.types.help*), 248  
[ProxyDataPromo](#) (*class in pyrogram.api.types.help*), 248  
[pyrogram.api.functions](#) (*module*), 143  
[pyrogram.api.functions.account](#) (*module*), 143  
[pyrogram.api.functions.auth](#) (*module*), 158  
[pyrogram.api.functions.bots](#) (*module*), 162  
[pyrogram.api.functions.channels](#) (*module*), 162  
[pyrogram.api.functions.contacts](#) (*module*), 171  
[pyrogram.api.functions.contest](#) (*module*), 175  
[pyrogram.api.functions.help](#) (*module*), 176  
[pyrogram.api.functions.langpack](#) (*module*), 180  
[pyrogram.api.functions.messages](#) (*module*), 182  
[pyrogram.api.functions.payments](#) (*module*), 217  
[pyrogram.api.functions.phone](#) (*module*), 219  
[pyrogram.api.functions.photos](#) (*module*), 221  
[pyrogram.api.functions.stickers](#) (*module*), 222  
[pyrogram.api.functions.updates](#) (*module*), 223  
[pyrogram.api.functions.upload](#) (*module*), 224  
[pyrogram.api.functions.users](#) (*module*), 227  
[pyrogram.api.types](#) (*module*), 232  
[pyrogram.api.types.account](#) (*module*), 232  
[pyrogram.api.types.auth](#) (*module*), 237  
[pyrogram.api.types.channels](#) (*module*), 240  
[pyrogram.api.types.contacts](#) (*module*), 241  
[pyrogram.api.types.help](#) (*module*), 245  
[pyrogram.api.types.messages](#) (*module*), 251  
[pyrogram.api.types.payments](#) (*module*), 264  
[pyrogram.api.types.phone](#) (*module*), 266  
[pyrogram.api.types.photos](#) (*module*), 267  
[pyrogram.api.types.storage](#) (*module*), 268  
[pyrogram.api.types.updates](#) (*module*), 270  
[pyrogram.api.types.upload](#) (*module*), 275  
[Pyrogrammer](#), 141
- ## R
- [RawUpdateHandler](#) (*class in pyrogram*), 100  
[ReadEncryptedHistory](#) (*class in pyrogram.api.functions.messages*), 202  
[ReadFeaturedStickers](#) (*class in pyrogram.api.functions.messages*), 202  
[ReadHistory](#) (*class in pyrogram.api.functions.channels*), 169

ReadHistory (class in *pyrogram.api.functions.messages*), 202  
 ReadMentions (class in *pyrogram.api.functions.messages*), 202  
 ReadMessageContents (class in *pyrogram.api.functions.channels*), 169  
 ReadMessageContents (class in *pyrogram.api.functions.messages*), 203  
 ReceivedCall (class in *pyrogram.api.functions.phone*), 220  
 ReceivedMessages (class in *pyrogram.api.functions.messages*), 203  
 ReceivedNotifyMessage (class in *pyrogram.api.types*), 402  
 ReceivedQueue (class in *pyrogram.api.functions.messages*), 203  
 RecentMeUrlChat (class in *pyrogram.api.types*), 403  
 RecentMeUrlChatInvite (class in *pyrogram.api.types*), 403  
 RecentMeUrls (class in *pyrogram.api.types.help*), 248  
 RecentMeUrlStickerSet (class in *pyrogram.api.types*), 403  
 RecentMeUrlUnknown (class in *pyrogram.api.types*), 403  
 RecentMeUrlUser (class in *pyrogram.api.types*), 404  
 RecentStickers (class in *pyrogram.api.types.messages*), 261  
 RecentStickersNotModified (class in *pyrogram.api.types.messages*), 261  
 RecoverPassword (class in *pyrogram.api.functions.auth*), 160  
 regex() (*pyrogram.Filters* static method), 106  
 RegisterDevice (class in *pyrogram.api.functions.account*), 150  
 remove\_cloud\_password() (*pyrogram.Client* method), 51  
 remove\_handler() (*pyrogram.Client* method), 23  
 RemoveStickerFromSet (class in *pyrogram.api.functions.stickers*), 223  
 ReorderPinnedDialogs (class in *pyrogram.api.functions.messages*), 203  
 ReorderStickerSets (class in *pyrogram.api.functions.messages*), 204  
 reply (*pyrogram.Filters* attribute), 104  
 reply() (*pyrogram.Message* method), 80  
 reply\_animation() (*pyrogram.Message* method), 81  
 reply\_audio() (*pyrogram.Message* method), 82  
 reply\_cached\_media() (*pyrogram.Message* method), 83  
 reply\_chat\_action() (*pyrogram.Message* method), 84  
 reply\_contact() (*pyrogram.Message* method), 84  
 reply\_document() (*pyrogram.Message* method), 85  
 reply\_game() (*pyrogram.Message* method), 86  
 reply\_inline\_bot\_result() (*pyrogram.Message* method), 87  
 reply\_keyboard (*pyrogram.Filters* attribute), 105  
 reply\_location() (*pyrogram.Message* method), 88  
 reply\_media\_group() (*pyrogram.Message* method), 88  
 reply\_photo() (*pyrogram.Message* method), 89  
 reply\_poll() (*pyrogram.Message* method), 90  
 reply\_sticker() (*pyrogram.Message* method), 91  
 reply\_venue() (*pyrogram.Message* method), 92  
 reply\_video() (*pyrogram.Message* method), 93  
 reply\_video\_note() (*pyrogram.Message* method), 94  
 reply\_voice() (*pyrogram.Message* method), 95  
 ReplyInlineMarkup (class in *pyrogram.api.types*), 404  
 ReplyKeyboardForceReply (class in *pyrogram.api.types*), 404  
 ReplyKeyboardHide (class in *pyrogram.api.types*), 404  
 ReplyKeyboardMarkup (class in *pyrogram*), 68  
 ReplyKeyboardMarkup (class in *pyrogram.api.types*), 404  
 ReplyKeyboardRemove (class in *pyrogram*), 69  
 Report (class in *pyrogram.api.functions.messages*), 204  
 ReportEncryptedSpam (class in *pyrogram.api.functions.messages*), 204  
 ReportPeer (class in *pyrogram.api.functions.account*), 151  
 ReportSpam (class in *pyrogram.api.functions.channels*), 170  
 ReportSpam (class in *pyrogram.api.functions.messages*), 204  
 ReqDHParams (class in *pyrogram.api.functions*), 231  
 ReqPq (class in *pyrogram.api.functions*), 231  
 ReqPqMulti (class in *pyrogram.api.functions*), 232  
 request\_callback\_answer() (*pyrogram.Client* method), 53  
 RequestCall (class in *pyrogram.api.functions.phone*), 220  
 RequestEncryption (class in *pyrogram.api.functions.messages*), 205  
 RequestPasswordRecovery (class in *pyrogram.api.functions.auth*), 160  
 ResendCode (class in *pyrogram.api.functions.auth*), 160  
 ResendPasswordEmail (class in *pyrogram.api.functions.account*), 151  
 ResetAuthorization (class in *pyrogram.api.functions.account*), 151  
 ResetAuthorizations (class in *pyrogram.api.functions.auth*), 160  
 ResetNotifySettings (class in *pyrogram*), 68

- [gram.api.functions.account](#)), 151  
[ResetSaved](#) (class in [pyrogram.api.functions.contacts](#)), 174  
[ResetTopPeerRating](#) (class in [pyrogram.api.functions.contacts](#)), 174  
[ResetWallPapers](#) (class in [pyrogram.api.functions.account](#)), 152  
[ResetWebAuthorization](#) (class in [pyrogram.api.functions.account](#)), 152  
[ResetWebAuthorizations](#) (class in [pyrogram.api.functions.account](#)), 152  
[resolve\\_peer\(\)](#) ([pyrogram.Client](#) method), 24  
[ResolvedPeer](#) (class in [pyrogram.api.types.contacts](#)), 244  
[ResolveUsername](#) (class in [pyrogram.api.functions.contacts](#)), 174  
[ResPQ](#) (class in [pyrogram.api.types](#)), 405  
[restart\(\)](#) ([pyrogram.Client](#) method), 23  
[restrict\\_chat\(\)](#) ([pyrogram.Client](#) method), 48  
[restrict\\_chat\\_member\(\)](#) ([pyrogram.Client](#) method), 42  
[retract\\_vote\(\)](#) ([pyrogram.Client](#) method), 40  
[ReuploadCdnFile](#) (class in [pyrogram.api.functions.upload](#)), 226  
[RPC](#), 141  
[RpcAnswerDropped](#) (class in [pyrogram.api.types](#)), 405  
[RpcAnswerDroppedRunning](#) (class in [pyrogram.api.types](#)), 405  
[RpcAnswerUnknown](#) (class in [pyrogram.api.types](#)), 406  
[RpcDropAnswer](#) (class in [pyrogram.api.functions](#)), 232  
[RPCError](#), 141  
[RpcError](#) (class in [pyrogram.api.types](#)), 406  
[RpcResult](#) (class in [pyrogram.api.types](#)), 406  
[run\(\)](#) ([pyrogram.Client](#) method), 23
- ## S
- [save\\_file\(\)](#) ([pyrogram.Client](#) method), 24  
[SaveAppLog](#) (class in [pyrogram.api.functions.help](#)), 180  
[SaveAutoDownloadSettings](#) (class in [pyrogram.api.functions.account](#)), 152  
[SaveBigFilePart](#) (class in [pyrogram.api.functions.upload](#)), 226  
[SaveCallDebug](#) (class in [pyrogram.api.functions.phone](#)), 221  
[SaveDeveloperInfo](#) (class in [pyrogram.api.functions.contest](#)), 175  
[SavedGifs](#) (class in [pyrogram.api.types.messages](#)), 261  
[SavedGifsNotModified](#) (class in [pyrogram.api.types.messages](#)), 262  
[SavedInfo](#) (class in [pyrogram.api.types.payments](#)), 266  
[SavedPhoneContact](#) (class in [pyrogram.api.types](#)), 406  
[SaveDraft](#) (class in [pyrogram.api.functions.messages](#)), 205  
[SaveFilePart](#) (class in [pyrogram.api.functions.upload](#)), 227  
[SaveGif](#) (class in [pyrogram.api.functions.messages](#)), 205  
[SaveRecentSticker](#) (class in [pyrogram.api.functions.messages](#)), 206  
[SaveSecureValue](#) (class in [pyrogram.api.functions.account](#)), 153  
[SaveWallPaper](#) (class in [pyrogram.api.functions.account](#)), 153  
[Search](#) (class in [pyrogram.api.functions.contacts](#)), 175  
[Search](#) (class in [pyrogram.api.functions.messages](#)), 206  
[SearchGifs](#) (class in [pyrogram.api.functions.messages](#)), 207  
[SearchGlobal](#) (class in [pyrogram.api.functions.messages](#)), 207  
[SearchStickerSets](#) (class in [pyrogram.api.functions.messages](#)), 207  
[SecureCredentialsEncrypted](#) (class in [pyrogram.api.types](#)), 407  
[SecureData](#) (class in [pyrogram.api.types](#)), 407  
[SecureFile](#) (class in [pyrogram.api.types](#)), 407  
[SecureFileEmpty](#) (class in [pyrogram.api.types](#)), 408  
[SecurePasswordKdfAlgoPBKDF2HMACSHA512iter100000](#) (class in [pyrogram.api.types](#)), 408  
[SecurePasswordKdfAlgoSHA512](#) (class in [pyrogram.api.types](#)), 408  
[SecurePasswordKdfAlgoUnknown](#) (class in [pyrogram.api.types](#)), 408  
[SecurePlainEmail](#) (class in [pyrogram.api.types](#)), 408  
[SecurePlainPhone](#) (class in [pyrogram.api.types](#)), 409  
[SecureRequiredType](#) (class in [pyrogram.api.types](#)), 409  
[SecureRequiredTypeOneOf](#) (class in [pyrogram.api.types](#)), 409  
[SecureSecretSettings](#) (class in [pyrogram.api.types](#)), 409  
[SecureValue](#) (class in [pyrogram.api.types](#)), 410  
[SecureValueError](#) (class in [pyrogram.api.types](#)), 410  
[SecureValueErrorData](#) (class in [pyrogram.api.types](#)), 411  
[SecureValueErrorFile](#) (class in [pyrogram.api.types](#)), 411  
[SecureValueErrorFiles](#) (class in [pyrogram.api.types](#)), 412



SecureValueErrorFrontSide (class in *pyrogram.api.types*), 412  
 SecureValueErrorReverseSide (class in *pyrogram.api.types*), 412  
 SecureValueErrorSelfie (class in *pyrogram.api.types*), 413  
 SecureValueErrorTranslationFile (class in *pyrogram.api.types*), 413  
 SecureValueErrorTranslationFiles (class in *pyrogram.api.types*), 414  
 SecureValueHash (class in *pyrogram.api.types*), 414  
 SecureValueTypeAddress (class in *pyrogram.api.types*), 414  
 SecureValueTypeBankStatement (class in *pyrogram.api.types*), 414  
 SecureValueTypeDriverLicense (class in *pyrogram.api.types*), 415  
 SecureValueTypeEmail (class in *pyrogram.api.types*), 415  
 SecureValueTypeIdentityCard (class in *pyrogram.api.types*), 415  
 SecureValueTypeInternalPassport (class in *pyrogram.api.types*), 415  
 SecureValueTypePassport (class in *pyrogram.api.types*), 415  
 SecureValueTypePassportRegistration (class in *pyrogram.api.types*), 416  
 SecureValueTypePersonalDetails (class in *pyrogram.api.types*), 416  
 SecureValueTypePhone (class in *pyrogram.api.types*), 416  
 SecureValueTypeRentalAgreement (class in *pyrogram.api.types*), 416  
 SecureValueTypeTemporaryRegistration (class in *pyrogram.api.types*), 416  
 SecureValueTypeUtilityBill (class in *pyrogram.api.types*), 417  
 send() (*pyrogram.Client* method), 23  
 send\_animation() (*pyrogram.Client* method), 30  
 send\_audio() (*pyrogram.Client* method), 27  
 send\_cached\_media() (*pyrogram.Client* method), 35  
 send\_chat\_action() (*pyrogram.Client* method), 36  
 send\_contact() (*pyrogram.Client* method), 35  
 send\_document() (*pyrogram.Client* method), 28  
 send\_game() (*pyrogram.Client* method), 54  
 send\_inline\_bot\_result() (*pyrogram.Client* method), 52  
 send\_location() (*pyrogram.Client* method), 34  
 send\_media\_group() (*pyrogram.Client* method), 33  
 send\_message() (*pyrogram.Client* method), 25  
 send\_photo() (*pyrogram.Client* method), 26  
 send\_poll() (*pyrogram.Client* method), 39  
 send\_sticker() (*pyrogram.Client* method), 29  
 send\_venue() (*pyrogram.Client* method), 34  
 send\_video() (*pyrogram.Client* method), 29  
 send\_video\_note() (*pyrogram.Client* method), 32  
 send\_voice() (*pyrogram.Client* method), 31  
 SendChangePhoneCode (class in *pyrogram.api.functions.account*), 153  
 SendCode (class in *pyrogram.api.functions.auth*), 161  
 SendConfirmPhoneCode (class in *pyrogram.api.functions.account*), 153  
 SendCustomRequest (class in *pyrogram.api.functions.bots*), 162  
 SendEncrypted (class in *pyrogram.api.functions.messages*), 208  
 SendEncryptedFile (class in *pyrogram.api.functions.messages*), 208  
 SendEncryptedService (class in *pyrogram.api.functions.messages*), 209  
 SendInlineBotResult (class in *pyrogram.api.functions.messages*), 209  
 SendMedia (class in *pyrogram.api.functions.messages*), 209  
 SendMessage (class in *pyrogram.api.functions.messages*), 210  
 SendMessageCancelAction (class in *pyrogram.api.types*), 417  
 SendMessageChooseContactAction (class in *pyrogram.api.types*), 417  
 SendMessageGamePlayAction (class in *pyrogram.api.types*), 417  
 SendMessageGeoLocationAction (class in *pyrogram.api.types*), 417  
 SendMessageRecordAudioAction (class in *pyrogram.api.types*), 418  
 SendMessageRecordRoundAction (class in *pyrogram.api.types*), 418  
 SendMessageRecordVideoAction (class in *pyrogram.api.types*), 418  
 SendMessageTypingAction (class in *pyrogram.api.types*), 418  
 SendMessageUploadAudioAction (class in *pyrogram.api.types*), 418  
 SendMessageUploadDocumentAction (class in *pyrogram.api.types*), 419  
 SendMessageUploadPhotoAction (class in *pyrogram.api.types*), 419  
 SendMessageUploadRoundAction (class in *pyrogram.api.types*), 419  
 SendMessageUploadVideoAction (class in *pyrogram.api.types*), 419  
 SendMultiMedia (class in *pyrogram.api.functions.messages*), 211  
 SendPaymentForm (class in *pyrogram.api.functions.payments*), 218  
 SendScreenshotNotification (class in *pyrogram*), 218

- gram.api.functions.messages*), 211
- SendVerifyEmailCode (class in *pyrogram.api.functions.account*), 154
- SendVerifyPhoneCode (class in *pyrogram.api.functions.account*), 154
- SendVote (class in *pyrogram.api.functions.messages*), 212
- SentCode (class in *pyrogram.api.types.auth*), 239
- SentCodeTypeApp (class in *pyrogram.api.types.auth*), 239
- SentCodeTypeCall (class in *pyrogram.api.types.auth*), 239
- SentCodeTypeFlashCall (class in *pyrogram.api.types.auth*), 240
- SentCodeTypeSms (class in *pyrogram.api.types.auth*), 240
- SentEmailCode (class in *pyrogram.api.types.account*), 235
- SentEncryptedFile (class in *pyrogram.api.types.messages*), 262
- SentEncryptedMessage (class in *pyrogram.api.types.messages*), 262
- ServerDHInnerData (class in *pyrogram.api.types*), 419
- ServerDHParamsFail (class in *pyrogram.api.types*), 420
- ServerDHParamsOk (class in *pyrogram.api.types*), 420
- service (*pyrogram.Filters* attribute), 105
- Session, 141
- set\_chat\_description() (*pyrogram.Client* method), 45
- set\_chat\_photo() (*pyrogram.Client* method), 44
- set\_chat\_title() (*pyrogram.Client* method), 45
- set\_game\_score() (*pyrogram.Client* method), 54
- set\_user\_profile\_photo() (*pyrogram.Client* method), 50
- SetAccountTTL (class in *pyrogram.api.functions.account*), 154
- SetBotCallbackAnswer (class in *pyrogram.api.functions.messages*), 212
- SetBotPrecheckoutResults (class in *pyrogram.api.functions.messages*), 212
- SetBotShippingResults (class in *pyrogram.api.functions.messages*), 213
- SetBotUpdatesStatus (class in *pyrogram.api.functions.help*), 180
- SetCallRating (class in *pyrogram.api.functions.phone*), 221
- SetClientDHParams (class in *pyrogram.api.functions*), 232
- SetContactSignUpNotification (class in *pyrogram.api.functions.account*), 154
- SetEncryptedTyping (class in *pyrogram.api.functions.messages*), 213
- SetGameScore (class in *pyrogram.api.functions.messages*), 213
- SetInlineBotResults (class in *pyrogram.api.functions.messages*), 214
- SetInlineGameScore (class in *pyrogram.api.functions.messages*), 214
- SetPrivacy (class in *pyrogram.api.functions.account*), 155
- SetSecureValueErrors (class in *pyrogram.api.functions.users*), 227
- SetStickers (class in *pyrogram.api.functions.channels*), 170
- SetTyping (class in *pyrogram.api.functions.messages*), 215
- ShippingOption (class in *pyrogram.api.types*), 420
- SignIn (class in *pyrogram.api.functions.auth*), 161
- SignUp (class in *pyrogram.api.functions.auth*), 161
- start() (*pyrogram.Client* method), 22
- StartBot (class in *pyrogram.api.functions.messages*), 215
- State (class in *pyrogram.api.types.updates*), 274
- StatsURL (class in *pyrogram.api.types*), 421
- Sticker (class in *pyrogram*), 67
- sticker (*pyrogram.Filters* attribute), 104
- StickerPack (class in *pyrogram.api.types*), 421
- Stickers (class in *pyrogram.api.types.messages*), 263
- StickerSet (class in *pyrogram.api.types*), 421
- StickerSet (class in *pyrogram.api.types.messages*), 263
- StickerSetCovered (class in *pyrogram.api.types*), 422
- StickerSetInstallResultArchive (class in *pyrogram.api.types.messages*), 263
- StickerSetInstallResultSuccess (class in *pyrogram.api.types.messages*), 263
- StickerSetMultiCovered (class in *pyrogram.api.types*), 422
- StickersNotModified (class in *pyrogram.api.types.messages*), 264
- stop() (*pyrogram.Client* method), 22
- stop\_poll() (*pyrogram.Client* method), 40
- stop\_transmission() (*pyrogram.Client* method), 25
- supergroup\_chat\_created (*pyrogram.Filters* attribute), 105
- Support (class in *pyrogram.api.types.help*), 249
- SupportName (class in *pyrogram.api.types.help*), 249

## T

- Takeout (class in *pyrogram.api.types.account*), 236
- TermsOfService (class in *pyrogram.api.types.help*), 249

TermsOfServiceUpdate (class in *pyrogram.api.types.help*), 250  
 TermsOfServiceUpdateEmpty (class in *pyrogram.api.types.help*), 250  
 text (*pyrogram.Filters* attribute), 104  
 TextAnchor (class in *pyrogram.api.types*), 422  
 TextBold (class in *pyrogram.api.types*), 423  
 TextConcat (class in *pyrogram.api.types*), 423  
 TextEmail (class in *pyrogram.api.types*), 423  
 TextEmpty (class in *pyrogram.api.types*), 423  
 TextFixed (class in *pyrogram.api.types*), 424  
 TextImage (class in *pyrogram.api.types*), 424  
 TextItalic (class in *pyrogram.api.types*), 424  
 TextMarked (class in *pyrogram.api.types*), 424  
 TextPhone (class in *pyrogram.api.types*), 425  
 TextPlain (class in *pyrogram.api.types*), 425  
 TextStrike (class in *pyrogram.api.types*), 425  
 TextSubscript (class in *pyrogram.api.types*), 425  
 TextSuperscript (class in *pyrogram.api.types*), 426  
 TextUnderline (class in *pyrogram.api.types*), 426  
 TextUrl (class in *pyrogram.api.types*), 426  
 TmpPassword (class in *pyrogram.api.types.account*), 236  
 ToggleDialogPin (class in *pyrogram.api.functions.messages*), 215  
 TogglePreHistoryHidden (class in *pyrogram.api.functions.channels*), 170  
 ToggleSignatures (class in *pyrogram.api.functions.channels*), 171  
 ToggleTopPeers (class in *pyrogram.api.functions.contacts*), 175  
 TopPeer (class in *pyrogram.api.types*), 426  
 TopPeerCategoryBotsInline (class in *pyrogram.api.types*), 427  
 TopPeerCategoryBotsPM (class in *pyrogram.api.types*), 427  
 TopPeerCategoryChannels (class in *pyrogram.api.types*), 427  
 TopPeerCategoryCorrespondents (class in *pyrogram.api.types*), 427  
 TopPeerCategoryGroups (class in *pyrogram.api.types*), 427  
 TopPeerCategoryPeers (class in *pyrogram.api.types*), 428  
 TopPeerCategoryPhoneCalls (class in *pyrogram.api.types*), 428  
 TopPeers (class in *pyrogram.api.types.contacts*), 244  
 TopPeersDisabled (class in *pyrogram.api.types.contacts*), 244  
 TopPeersNotModified (class in *pyrogram.api.types.contacts*), 245  
 Unblock (class in *pyrogram.api.functions.contacts*), 175  
 UninstallStickerSet (class in *pyrogram.api.functions.messages*), 216  
 unpin\_chat\_message () (*pyrogram.Client* method), 46  
 UnregisterDevice (class in *pyrogram.api.functions.account*), 155  
 Update (class in *pyrogram.api.types*), 428  
 update\_chat\_username () (*pyrogram.Client* method), 49  
 update\_username () (*pyrogram.Client* method), 50  
 UpdateBotCallbackQuery (class in *pyrogram.api.types*), 429  
 UpdateBotInlineQuery (class in *pyrogram.api.types*), 430  
 UpdateBotInlineSend (class in *pyrogram.api.types*), 430  
 UpdateBotPrecheckoutQuery (class in *pyrogram.api.types*), 430  
 UpdateBotShippingQuery (class in *pyrogram.api.types*), 431  
 UpdateBotWebhookJSON (class in *pyrogram.api.types*), 431  
 UpdateBotWebhookJSONQuery (class in *pyrogram.api.types*), 431  
 UpdateChannel (class in *pyrogram.api.types*), 431  
 UpdateChannelAvailableMessages (class in *pyrogram.api.types*), 432  
 UpdateChannelMessageViews (class in *pyrogram.api.types*), 432  
 UpdateChannelPinnedMessage (class in *pyrogram.api.types*), 432  
 UpdateChannelReadMessagesContents (class in *pyrogram.api.types*), 432  
 UpdateChannelTooLong (class in *pyrogram.api.types*), 433  
 UpdateChannelWebPage (class in *pyrogram.api.types*), 433  
 UpdateChatDefaultBannedRights (class in *pyrogram.api.types*), 433  
 UpdateChatParticipantAdd (class in *pyrogram.api.types*), 434  
 UpdateChatParticipantAdmin (class in *pyrogram.api.types*), 434  
 UpdateChatParticipantDelete (class in *pyrogram.api.types*), 434  
 UpdateChatParticipants (class in *pyrogram.api.types*), 435  
 UpdateChatPinnedMessage (class in *pyrogram.api.types*), 435  
 UpdateChatUserTyping (class in *pyrogram.api.types*), 435  
 unban\_chat\_member () (*pyrogram.Client* method), 42

## U

`UpdateConfig` (class in `pyrogram.api.types`), 435

`UpdateContactLink` (class in `pyrogram.api.types`), 436

`UpdateContactsReset` (class in `pyrogram.api.types`), 436

`UpdateDcOptions` (class in `pyrogram.api.types`), 436

`UpdateDeleteChannelMessages` (class in `pyrogram.api.types`), 436

`UpdateDeleteMessages` (class in `pyrogram.api.types`), 437

`UpdateDeviceLocked` (class in `pyrogram.api.functions.account`), 155

`UpdateDialogPinned` (class in `pyrogram.api.types`), 437

`UpdateDialogUnreadMark` (class in `pyrogram.api.types`), 437

`UpdateDraftMessage` (class in `pyrogram.api.types`), 437

`UpdateEditChannelMessage` (class in `pyrogram.api.types`), 438

`UpdateEditMessage` (class in `pyrogram.api.types`), 438

`UpdateEncryptedChatTyping` (class in `pyrogram.api.types`), 438

`UpdateEncryptedMessagesRead` (class in `pyrogram.api.types`), 438

`UpdateEncryption` (class in `pyrogram.api.types`), 439

`UpdateFavedStickers` (class in `pyrogram.api.types`), 439

`UpdateInlineBotCallbackQuery` (class in `pyrogram.api.types`), 439

`UpdateLangPack` (class in `pyrogram.api.types`), 439

`UpdateLangPackTooLong` (class in `pyrogram.api.types`), 440

`UpdateMessageID` (class in `pyrogram.api.types`), 440

`UpdateMessagePoll` (class in `pyrogram.api.types`), 440

`UpdateNewChannelMessage` (class in `pyrogram.api.types`), 440

`UpdateNewEncryptedMessage` (class in `pyrogram.api.types`), 441

`UpdateNewMessage` (class in `pyrogram.api.types`), 441

`UpdateNewStickerSet` (class in `pyrogram.api.types`), 441

`UpdateNotifySettings` (class in `pyrogram.api.functions.account`), 155

`UpdateNotifySettings` (class in `pyrogram.api.types`), 441

`UpdatePasswordSettings` (class in `pyrogram.api.functions.account`), 156

`UpdatePhoneCall` (class in `pyrogram.api.types`), 442

`UpdatePinnedDialogs` (class in `pyrogram.api.types`), 442

`UpdatePinnedMessage` (class in `pyrogram.api.functions.messages`), 216

`UpdatePrivacy` (class in `pyrogram.api.types`), 442

`UpdateProfile` (class in `pyrogram.api.functions.account`), 156

`UpdateProfilePhoto` (class in `pyrogram.api.functions.photos`), 222

`UpdatePtsChanged` (class in `pyrogram.api.types`), 442

`UpdateReadChannelInbox` (class in `pyrogram.api.types`), 443

`UpdateReadChannelOutbox` (class in `pyrogram.api.types`), 443

`UpdateReadFeaturedStickers` (class in `pyrogram.api.types`), 443

`UpdateReadHistoryInbox` (class in `pyrogram.api.types`), 443

`UpdateReadHistoryOutbox` (class in `pyrogram.api.types`), 444

`UpdateReadMessagesContents` (class in `pyrogram.api.types`), 444

`UpdateRecentStickers` (class in `pyrogram.api.types`), 444

`UpdateSavedGifs` (class in `pyrogram.api.types`), 444

`UpdatesCombined` (class in `pyrogram.api.types`), 452

`UpdateServiceNotification` (class in `pyrogram.api.types`), 445

`UpdateShort` (class in `pyrogram.api.types`), 445

`UpdateShortChatMessage` (class in `pyrogram.api.types`), 446

`UpdateShortMessage` (class in `pyrogram.api.types`), 447

`UpdateShortSentMessage` (class in `pyrogram.api.types`), 448

`UpdateStatus` (class in `pyrogram.api.functions.account`), 156

`UpdateStickerSets` (class in `pyrogram.api.types`), 449

`UpdateStickerSetsOrder` (class in `pyrogram.api.types`), 449

`UpdatesTooLong` (class in `pyrogram.api.types`), 453

`UpdateUserBlocked` (class in `pyrogram.api.types`), 449

`UpdateUsername` (class in `pyrogram.api.functions.account`), 157

`UpdateUsername` (class in `pyrogram.api.functions.channels`), 171

`UpdateUserName` (class in `pyrogram.api.types`), 450

`UpdateUserPhone` (class in `pyrogram.api.types`), 450

`UpdateUserPhoto` (class in `pyrogram.api.types`), 450

`UpdateUserPinnedMessage` (class in `pyrogram.api.types`), 451

`UpdateUserStatus` (class in `pyrogram.api.types`),



451  
 UpdateUserTyping (class in *pyrogram.api.types*), 451  
 UpdateWebPage (class in *pyrogram.api.types*), 451  
 UploadEncryptedFile (class in *pyrogram.api.functions.messages*), 216  
 UploadMedia (class in *pyrogram.api.functions.messages*), 217  
 UploadProfilePhoto (class in *pyrogram.api.functions.photos*), 222  
 UploadWallPaper (class in *pyrogram.api.functions.account*), 157  
 User (class in *pyrogram*), 57  
 User (class in *pyrogram.api.types*), 453  
 Userbot, 141  
 UserEmpty (class in *pyrogram.api.types*), 454  
 UserFull (class in *pyrogram.api.types*), 455  
 UserInfo (class in *pyrogram.api.types.help*), 250  
 UserInfoEmpty (class in *pyrogram.api.types.help*), 251  
 UserProfilePhoto (class in *pyrogram.api.types*), 455  
 UserProfilePhotoEmpty (class in *pyrogram.api.types*), 456  
 UserProfilePhotos (class in *pyrogram*), 65  
 UserStatus (class in *pyrogram*), 57  
 UserStatusEmpty (class in *pyrogram.api.types*), 456  
 UserStatusHandler (class in *pyrogram*), 100  
 UserStatusLastMonth (class in *pyrogram.api.types*), 456  
 UserStatusLastWeek (class in *pyrogram.api.types*), 456  
 UserStatusOffline (class in *pyrogram.api.types*), 456  
 UserStatusOnline (class in *pyrogram.api.types*), 457  
 UserStatusRecently (class in *pyrogram.api.types*), 457

## V

ValidatedRequestedInfo (class in *pyrogram.api.types.payments*), 266  
 ValidateRequestedInfo (class in *pyrogram.api.functions.payments*), 218  
 Venue (class in *pyrogram*), 67  
 venue (*pyrogram.Filters* attribute), 104  
 VerifyEmail (class in *pyrogram.api.functions.account*), 157  
 VerifyPhone (class in *pyrogram.api.functions.account*), 157  
 via\_bot (*pyrogram.Filters* attribute), 105  
 Video (class in *pyrogram*), 66  
 video (*pyrogram.Filters* attribute), 104  
 video\_note (*pyrogram.Filters* attribute), 104

VideoNote (class in *pyrogram*), 66  
 Voice (class in *pyrogram*), 66  
 voice (*pyrogram.Filters* attribute), 104  
 vote\_poll () (*pyrogram.Client* method), 40

## W

WallPaper (class in *pyrogram.api.types*), 457  
 WallPapers (class in *pyrogram.api.types.account*), 236  
 WallpaperSettings (class in *pyrogram.api.types*), 457  
 WallPapersNotModified (class in *pyrogram.api.types.account*), 236  
 web\_page (*pyrogram.Filters* attribute), 104  
 WebAuthorization (class in *pyrogram.api.types*), 458  
 WebAuthorizations (class in *pyrogram.api.types.account*), 237  
 WebDocument (class in *pyrogram.api.types*), 458  
 WebDocumentNoProxy (class in *pyrogram.api.types*), 459  
 WebFile (class in *pyrogram.api.types.upload*), 276  
 WebPage (class in *pyrogram.api.types*), 459  
 WebPageEmpty (class in *pyrogram.api.types*), 460  
 WebPageNotModified (class in *pyrogram.api.types*), 460  
 WebPagePending (class in *pyrogram.api.types*), 460