

./ instagrapi

🔥 The fastest and powerful Python library for Instagram
Private API 2024

[View on GitHub](#)

instagrapi

We recommend using our services:

>> [LamaTok](#) for TikTok API 🔥
>> [HikerAPI](#) for Instagram API ⚡⚡⚡
>> [DataLikers](#) for Instagram Datasets 🚀

📦 Package **failing** pypi v2.1.2 python 3.9 | 3.10

Fast and effective Instagram Private API wrapper (public+private requests and challenge resolver). Use the most recent version of the API from Instagram, which was obtained using [reverse-engineering with Charles Proxy](#) and [Proxyman](#).

Instagram API valid for 16 Dec 2023 (last reverse-engineering check)

Support Python >= 3.9

For any other languages (e.g. C++, C#, F#, D, [Golang](#), Erlang, Elixir, Nim, Haskell, Lisp, Closure, Julia, R, Java, Kotlin, Scala, OCaml, JavaScript, Crystal, Ruby, Rust, [Swift](#), Objective-C, Visual Basic, .NET, Pascal, Perl, Lua, PHP and others), I suggest using [instagrapi-rest](#)

[Support Chat in Telegram](#) 📬 and [GitHub Discussions](#)

Features

1. Performs [Public API](#) (web, anonymous) or [Private API](#) (mobile app, authorized) requests depending on the situation (to avoid Instagram limits)
2. [Login](#) by username and password, including 2FA and by sessionid

3. [Challenge Resolver](#) have Email and SMS handlers
4. Support [upload](#) a Photo, Video, IGTV, Reels, Albums and Stories
5. Support work with [User](#), [Media](#), [Comment](#), [Insights](#), [Collections](#), [Location](#) (Place), [Hashtag](#) and [Direct Message](#) objects
6. [Like](#), [Follow](#), [Edit account](#) (Bio) and much more else
7. [Insights](#) by account, posts and stories
8. [Build stories](#) with custom background, font animation, swipe up link and mention users
9. In the next release, account registration and captcha passing will appear

Example

Basic Usage

```
from instagrapi import Client

cl = Client()
cl.login(ACCOUNT_USERNAME, ACCOUNT_PASSWORD)

user_id = cl.user_id_from_username("example")
medias = cl.user_medias(user_id, 20)
```

The full example

```
from instagrapi import Client
from instagrapi.types import StoryMention, StoryMedia, StoryLink, StoryHashtag

cl = Client()
cl.login(USERNAME, PASSWORD, verification_code="<2FA CODE HERE>")

media_pk = cl.media_pk_from_url('https://www.instagram.com/p/CGgDsi7JQdS')
media_path = cl.video_download(media_pk)
example = cl.user_info_by_username('example')
hashtag = cl.hashtag_info('dhabastards')

cl.video_upload_to_story(
    media_path,
    "Credits @example",
    mentions=[StoryMention(user=example, x=0.49892962, y=0.703125, width=0.2, height=0.2)],
    links=[StoryLink(webUri='https://github.com/subzeroid/instagrapi')],
    hashtags=[StoryHashtag(hashtag=hashtag, x=0.23, y=0.32, width=0.5, height=0.2)]
)
```

```
medias=[StoryMedia(media_pk=media_pk, x=0.5, y=0.5, width=0.6, height=0.6)
)
```

Requests

- >> Public (anonymous request via web api) methods have a suffix `_gql` (Instagram GraphQL) or `_a1` (example `https://www.instagram.com/example/?__a=1`)
- >> Private (authorized request via mobile api) methods have `_v1` suffix

The first request to fetch media/user is `public` (anonymous), if `instagram` raise exception, then use `private` (authorized). Example (pseudo-code):

```
def media_info(media_pk):
    try:
        return self.media_info_gql(media_pk)
    except ClientError as e:
        # Restricted Video: This video is not available in your country.
        # Or media from private account
        return self.media_info_v1(media_pk)
```

Detailed Documentation

To learn more about the various ways `instagrapi` can be used, read the [Usage Guide](#) page.

- >> [Getting Started](#)
- >> [Usage Guide](#)
- >> [Interactions](#)
 - >> [Media](#) - Publication (also called post): Photo, video, Album, IGTV and Reels
 - >> [Resource](#) - Part of Media (for albums)
 - >> [MediaOembed](#) - Short version of Media
 - >> [Account](#) - Full private info for your account (e.g. email, phone_number)
 - >> [TOTP](#) - 2FA TOTP helpers (generate seed, enable/disable TOTP, generate code as Google Authenticator)
 - >> [User](#) - Full public user data
 - >> [UserShort](#) - Short public user data (used in Usertag, Comment, Media, Direct Message)
 - >> [Usertag](#) - Tag user in Media (coordinates + UserShort)
 - >> [Location](#) - GEO location (GEO coordinates, name, address)
 - >> [Hashtag](#) - Hashtag object (id, name, picture)

- >> [Collection](#) - Collection of medias (name, picture and list of medias)
- >> [Comment](#) - Comments to Media
- >> [Highlight](#) - Highlights
- >> [‘Notes’](#) - Notes
- >> [Story](#) - Story
- >> [StoryLink](#) - Link (Swipe up)
- >> [StoryLocation](#) - Tag Location in Story (as sticker)
- >> [StoryMention](#) - Mention users in Story (user, coordinates and dimensions)
- >> [StoryHashtag](#) - Hashtag for story (as sticker)
- >> [StorySticker](#) - Tag sticker to story (for example from giphy)
- >> [StoryBuild](#) - [StoryBuilder](#) return path to photo/video and mention co-ordinates
- >> [DirectThread](#) - Thread (topic) with messages in Direct Message
- >> [DirectMessage](#) - Message in Direct Message
- >> [Insight](#) - Insights for a post
- >> [Track](#) - Music track (for Reels/Clips)
- >> [Best Practices](#)
- >> [Development Guide](#)
- >> [Handle Exceptions](#)
- >> [Challenge Resolver](#)
- >> [Exceptions](#)