
Instauto

Release 2.0.6

Stan van Rooy

Jan 24, 2023

CONTENTS

1	Usage	3
	Python Module Index	23
	Index	25

A Python wrapper for the private Instagram API.

USAGE

Instauto has 3 main api's that can be used: *instauto.api*, *instauto.bot* & *instauto.helpers*.

Everything in *instauto* is based around the 'core' *instauto.api* package. This package interacts directly with the private Instagram API and contains all core functionality. This package is both the most flexible (you can update all requests sent and receive the full response back, for example), but also the most complex.

The *instauto.helpers* package is an abstraction above the *instauto.api* package. It offers less flexibility, but is simpler to use.

The *instauto.bot* package is another abstraction above the *instauto.helpers* package. This package has pretty much no flexibility, but can be set up in 10 lines of Python code.

1.1 instauto.api package

1.1.1 instauto.api.actions package

instauto.api.actions.authentication module

class `instauto.api.actions.authentication.AuthenticationMixin`

Bases: `StubMixin`

change_password(*new_password: str*, *current_password: Optional[str] = None*) → Response

device_profile: `DeviceProfile`

ig_profile: `IGProfile`

log_in() → None

Logs in the account with the *username* and *password*

state: `State`

instauto.api.actions.challenge module**class** instauto.api.actions.challenge.ChallengeMixinBases: *StubMixin***device_profile:** *DeviceProfile***ig_profile:** *IGProfile***state:** *State***instauto.api.actions.direct module****class** instauto.api.actions.direct.DirectMixinBases: *StubMixin***device_profile:** *DeviceProfile***direct_get_thread**(obj: *DirectThread*) → *Thread*

Retrieve more information about a thread.

If this thread exists in the inbox, it will be updated. If not, it will be added to the thread lists.

Returns

The retrieved thread.

Return type*Thread***direct_send**(obj: *Union[Message, MediaShare, LinkShare, ProfileShare, DirectPhoto, DirectVideo]*) →*Response*

Send a message to a thread.

direct_update_inbox() → bool

Request your inbox status from Instagram.

Updates the threads with a distinct set of the old & new threads. Overwrites all other properties.

Returns

True if the inbox has been updated

Return type

bool

ig_profile: *IGProfile***property** inbox**state:** *State*

instauto.api.actions.friendships module

class instauto.api.actions.friendships.**FriendshipsMixin**

Bases: *StubMixin*

device_profile: *DeviceProfile*

follow_request_approve(*obj: ApproveRequest*) → Response

Accept a follow request/

follow_requests_get(*obj: PendingRequests*) → List[dict]

Retrieve all follow requests

follower_remove(*obj: Remove*) → Response

Remove someone from your followers list, that is currently following you

follower_show(*obj: Show*) → Response

Retrieve information about a user

followers_get(*obj: GetFollowers*) → Tuple[GetFollowers, Union[Response, bool]]

Retrieves the followers of an Instagram user.

Returns

(GetFollowers, Response || bool): A tuple that contains the object that was passed in as an argument, but with updated max_id and page attributes, and the response or False. If the second item is False, there were no more items available.

following_get(*obj: GetFollowing*) → Tuple[GetFollowing, Union[Response, bool]]

Retrieves the following of an Instagram user.

Returns

A tuple that contains the object that was passed in as an argument, but with updated max_id and page attributes, and the response or False. If the second item is False, there were no more items available.

Return type

(GetFollowing, Response || bool)

ig_profile: *IGProfile*

state: *State*

user_follow(*obj: Create*) → Response

Follow a user

user_unfollow(*obj: Destroy*) → Response

Unfollow a user

instauto.api.actions.helpers module

class instauto.api.actions.helpers.**HelperMixin**

Bases: *StubMixin*

device_profile: *DeviceProfile*

static get_image_type(*p: Union[str, Path]*) → str

Returns the type of image, i.e. jpeg or png.

ig_profile: *IGProfile*

state: *State*

instauto.api.actions.post module

class instauto.api.actions.post.**PostMixin**

Bases: *StubMixin*

device_profile: *DeviceProfile*

ig_profile: *IGProfile*

post_archive(*obj: Archive*) → Response

post_carousel(*posts: List[PostFeed], caption: str, quality: int*) → Dict[str, Response]

post_comment(*obj: Comment*) → Response

Comments on a post

post_get_commenters(*obj: RetrieveCommenters*) → List[Dict]

post_get_comments(*obj: RetrieveComments*) → Response

post_get_likers(*obj: RetrieveLikers*) → List[Dict]

Retrieve all likers of specific media_id

post_like(*obj: Like*) → Response

Likes a post

post_post(*obj: Union[PostStory, PostFeed, PostNull], quality: Optional[int] = None*) → Response

Uploads a new picture/video to your Instagram account. :param obj: Should be instantiated with all the required params :type obj: Post :param quality: Quality of the image, defaults to 70. :type quality: int

Returns

The response returned by the Instagram API.

Return type

Response

post_retrieve_by_id(*obj: RetrieveById*) → Response

post_retrieve_by_tag(*obj: RetrieveByTag*) → Tuple[RetrieveByTag, Union[dict, bool]]

post_retrieve_by_user(*obj: RetrieveByUser*) → Tuple[RetrieveByUser, Union[dict, bool]]

Retrieves 12 posts of the user at a time. If there was a response / if there were any more posts available, the response can be found in original_requests/post.json:4

Returns

Will return the updated object and the response if there were any posts left, returns the object and False if not.

Return type

PostRetrieveByUser, (dict, bool)

post_retrieve_story(*obj: RetrieveStory*) → Response

post_save(*obj: Save*) → Response

Saves a post to your Instagram account

post_unarchive(*obj: Unarchive*) → Response

post_unlike(*obj: Unlike*) → Response
Unlikes a post

post_update_caption(*obj: UpdateCaption*) → Response
Updates the caption of a post

state: *State*

instauto.api.actions.profile module

class instauto.api.actions.profile.**ProfileMixin**
Bases: *StubMixin*

device_profile: *DeviceProfile*

ig_profile: *IGProfile*

profile_info(*obj: Info*) → Union[Dict, int]

profile_set_biography(*obj: SetBiography*) → Response
Sets the biography of the currently logged in user

profile_set_gender(*obj: SetGender*) → Response
Sets the gender of the currently logged in user

profile_set_picture(*obj: SetPicture*) → Response

profile_update(*obj: Update*)
Updates the name, username, email, phone number and url for the currently logged in user.

state: *State*

instauto.api.actions.request module

class instauto.api.actions.request.**RequestMixin**
Bases: *StubMixin*

device_profile: *DeviceProfile*

ig_profile: *IGProfile*

state: *State*

instauto.api.actions.search module

class instauto.api.actions.search.**SearchMixin**
Bases: *StubMixin*

device_profile: *DeviceProfile*

ig_profile: *IGProfile*

search_tag(*obj: Tag*) → Response

`search_username(obj: Username) → Response`

`state: State`

instauto.api.actions.stub module

`class instauto.api.actions.stub.StubMixin`

Bases: `object`

`device_profile: DeviceProfile`

`ig_profile: IGProfile`

`state: State`

1.1.2 instauto.api.client module

`class instauto.api.client.ApiClient(ig_profile: Optional[IGProfile] = None, device_profile: Optional[DeviceProfile] = None, state: Optional[State] = None, username: Optional[str] = None, password: Optional[str] = None, session_cookies: Optional[dict] = None, testing=False, _2fa_function: Optional[Callable[[str], str]] = None)`

Bases: `ProfileMixin, AuthenticationMixin, PostMixin, RequestMixin, FriendshipsMixin, SearchMixin, ChallengeMixin, DirectMixin, HelperMixin, FeedMixin, ActivityMixin`

`bc_hmac = <hmac.HMAC object>`

`breadcrumb_private_key = b'iN4$aGr0m'`

`device_profile: DeviceProfile`

`classmethod from_json(j: Union[str, bytes]) → ApiClient`

`ig_profile: IGProfile`

`classmethod initiate_from_file(file_name: str) → ApiClient`

`save_to_disk(file_name: str, overwrite: bool = False) → bool`

`state: State`

`to_json() → str`

1.1.3 instauto.api.constants module

`instauto.api.constants.DEFAULT_SIGNATURE_KEY = '19ce5f445dbfd9d29c59dc2a78c616a7fc090a8e018b9267bc4240a30244c53b'`

Can change overtime, but it's pretty easy to extract, see: <https://mokhdzanifaeq.github.io/2015/09/28/extracting-instagram-signature-key-2/>

1.1.4 instauto.api.exceptions module

exception instauto.api.exceptions.**AuthorizationError**

Bases: Exception

Raised when you try to get an object you're not authorized to get

exception instauto.api.exceptions.**BadResponse**

Bases: Exception

Raised when Instagram returns a non-ok status code.

exception instauto.api.exceptions.**CorruptedSaveData**

Bases: Exception

Raised when the save data can't be read

exception instauto.api.exceptions.**IncorrectLoginDetails**

Bases: Exception

Raised when the provided logging details are incorrect.

exception instauto.api.exceptions.**InvalidUserId**

Bases: Exception

Raised when an invalid user id is provided

exception instauto.api.exceptions.**MissingValue**

Bases: Exception

Raised when an action struct is initiated with a missing value

exception instauto.api.exceptions.**NoAuthDetailsProvided**

Bases: Exception

Raised when the login details are not provided, but the client needs them

exception instauto.api.exceptions.**NotFoundError**

Bases: Exception

Raised when an entity is not found.

exception instauto.api.exceptions.**StateExpired**

Bases: Exception

Raised when saved settings are provided, but not valid anymore

1.1.5 instauto.api.structs module

class instauto.api.structs.**ChallengeChoice**(value)

Bases: Enum

An enumeration.

email = 1

phone = 0

```
class instauto.api.structs.DeviceProfile(manufacturer: Optional[str] = None, android_sdk_version:  
Optional[str] = None, android_release: Optional[str] = None,  
device: Optional[str] = None, model: Optional[str] = None,  
dpi: Optional[int] = None, resolution: Optional[Tuple[int]] =  
None, chipset: Optional[str] = None)
```

Bases: object

Holds all data about the android 'phone' that we simulate using.

manufacturer

The phone manufacturer

Type

str,

android_sdk_version

The Android sdk version that is, presumably, used by the Instagram app.

Type

str,

android_release

The version of Android that the phone runs on.

Type

str,

device

The version name of the phone

Type

str,

model

The codename from Samsung under which the phone was build, i.e. for the Galaxy S10E, beyond1.

Type

str,

dpi

The DPI of the phone used.

Type

str,

resolution

The resolution of the phone.

Type

tuple[int, int],

chipset

The chipset that the phone runs on.

```
class instauto.api.structs.IGProfile(signature_key: Optional[str] = None, signature_key_version:  
Optional[str] = None, http_engine: Optional[str] = None,  
capabilities: Optional[str] = None, id: Optional[str] = None,  
version: Optional[str] = None, build_number: Optional[str] =  
None)
```

Bases: object

Holds all data that is generated by Instagram. For pretty much every request, at least one of the attributes is used.

signature_key

Key generated by Instagram to sign post requests. Can be extracted from the app. Currently, the actual signature key is no longer used for signing actual requests.

Type

str, DEPRECATED

signature_key

The version of the signature key. This key is still sent along with signed requests. Could probably work without. TODO: check if we still need to send this along with signed requests / if we have to use the signed request format at all

Type

str

http_engine

Facebook uses a custom HTTP engine, called Liger. This is unlikely to change.

Type

str,

capabilities

Not sure what this means on Instagram's side, but it needs to get sent along with all requests. Can change overtime. Can be extracted from all requests to the 'logging_client_events' endpoint.

Type

str,

id

The app id, presumably a constant.

Type

str,

version

The version number of the version of instagram to use.

Type

str,

build_number

The build number associated with the version number

Type

str,

```
class instauto.api.structs.Inbox(threads: List[instauto.api.structs.Thread], has_older: bool, unseen_count:
    int, unseen_count_ts: int, oldest_cursor: str, prev_cursor: dict,
    next_cursor: dict, blended_inbox_enabled: bool, seq_id: int,
    snapshot_at_ms: int, pending_requests_total: int,
    has_pending_top_requests: bool)
```

Bases: object

blended_inbox_enabled: `bool`
True if blended inbox is available, else False

has_older: `bool`
True if there are older threads available, else False

has_pending_top_requests: `bool`
True if inbox has pending top requests else False

next_cursor: `dict`
ID of the next cursor

oldest_cursor: `str`
ID of the oldest cursor

pending_requests_total: `int`
Count of pending message requests.

prev_cursor: `dict`
ID of the previous cursor

seq_id: `int`
Sequence identifier. Used by Instagram to track which messages have been received.

snapshot_at_ms: `int`
Timestamp of when this snapshot of the inbox was taken

threads: `List[Thread]`
a list that contains all your threads(chats)

unseen_count: `int`
The amount of threads with unseen threads

unseen_count_ts: `int`
Timestamp of the last check for unseen threads


```

class instauto.api.structs.LoggedInAccountData(account_type: Optional[int] = None, account_badges:
Optional[list] = None, allow_contacts_sync:
Optional[bool] = None, allowed_commenter_type:
Optional[str] = None, can_boost_post: Optional[bool] =
None, can_see_organic_insights: Optional[bool] =
None, can_see_primary_country_in_settings:
Optional[bool] = None, full_name: Optional[str] =
None, has_anonymous_profile_picture: Optional[bool] =
None, has_placed_orders: Optional[bool] = None,
interop_messaging_user_fbid: Optional[int] = None,
is_business: Optional[bool] = None,
is_call_to_action_enabled: Optional[bool] = None,
nametag: Optional[dict] = None, phone_number:
Optional[str] = None, pk: Optional[int] = None,
professional_conversion_suggested_account_type:
Optional[int] = None, profile_pic_id: Optional[str] =
None, profile_pic_url: Optional[str] = None,
show_insights_terms: Optional[bool] = None,
total_igtv_videos: Optional[int] = None, username:
Optional[str] = None, is_private: Optional[bool] =
None, is_verified: Optional[bool] = None,
reel_auto_archive: Optional[str] = None,
is_using_unified_inbox_for_direct: Optional[bool] =
None, can_hide_category: Optional[str] = None,
can_hide_public_contacts: Optional[str] = None,
*args, **kwargs)

```

Bases: object

Structure that stores information about the Instagram account

```
class instauto.api.structs.Method(value)
```

Bases: Enum

An enumeration.

GET = 1

POST = 2

```
class instauto.api.structs.PostLocation(value)
```

Bases: Enum

An enumeration.

Feed = 4

Post to feed

Null = 0

Post without configuring

Story = 3

Post to story

```
class instauto.api.structs.State(app_startup_country: Optional[str] = None, device_locale: Optional[str]
                                = None, app_locale: Optional[str] = None, bandwidth_totalbytes_b:
                                Optional[str] = None, bandwidth_totaltime_ms: Optional[str] = None,
                                connection_type: Optional[str] = None, accept_language: Optional[str]
                                = None, accept_encoding: Optional[str] = None, accept: Optional[str] =
                                None, ads_opt_out: Optional[bool] = None, authorization: Optional[str]
                                = None, www_claim: Optional[str] = None, rur: Optional[str] = None,
                                blos_version_id: Optional[str] = None, blos_is_layout_rtl:
                                Optional[str] = None, battery_level: Optional[int] = None, is_charging:
                                Optional[int] = None, is_dark_mode: Optional[bool] = None, **kwargs)
```

Bases: object

Structure that holds a lot of data about the state of a session. It contains mainly header values that need to be send along with requests to the API.

www_claim

Some sort of tracking / identifying header value that is send along with every HTTP request. It is also updated in almost all responses received from Instagram's API.

Type

str,

authorization

Contains the token used for Bearer authentication.

Type

str,

mid

Another tracking / identifying header value. Is also sent along with all requests. Is also updated in every response.

Type

str,

logged_in_account_data

Gets filled as soon as you login. Contains a lot of data about your account.

Type

LoggedInAccountData,

property android_id

Creates an Android id from the device id.

property bandwidth_speed_kbps

Randomizes the bandwidth speed

property connection_speed: str

Randomizes the connection speed.

fill(f: Callable) → None

Fills the State with initial values.

Initializes all variables that:

- 1) do not have a default value to start with;
- 2) need a unique generated key on a per-user basis

Parameters

f (*function*) – The function that generates the unique keys used throughout.

refresh(*f: Callable*)

Refresh the state. Simulates a re-open of the Instagram app.

property startup_country: **str**

property valid: **bool**

Sessions older then 90 days will not work anymore.

class instauto.api.structs.**Surface**(*value*)

Bases: Enum

An enumeration.

follow_list = 'follow_list_page'

follow_requests = 'follow_requests'

following_list = 'self_unified_follow_lists'

profile = 'following_sheet'

class instauto.api.structs.**Thread**(*thread_id: str, thread_v2_id: str, users: List[dict], left_users: List[dict], admin_user_ids: List[dict], items: List[dict], properties: dict*)

Bases: object

admin_user_ids: **List[dict]**

users that are admins in the thread

items: **List[dict]**

a list of all messages sent in the thread

left_users: **List[dict]**

users that have left the thread

properties: **dict**

all other properties

thread_id: **str**

thread_v2_id: **str**

users: **List[dict]**

class instauto.api.structs.**WhichGender**(*value*)

Bases: Enum

An enumeration.

female = 2

male = 1

other = 4

prefer_not_to_say = 3

1.2 instauto.helpers package

1.2.1 instauto.helpers.friendships module

`instauto.helpers.friendships.follow_user(client: ApiClient, user_id: Optional[int] = None, username: Optional[str] = None) → bool`

Send a follow request to a user.

Either *user_id* or *username* need to be provided. If both are provided, the *user_id* takes precedence.

Parameters

- **client** – your ApiClient
- **user_id** – the user_id of the account to follow
- **username** – the username of the account to follow

Returns

True if success else False

`instauto.helpers.friendships.get_followers(client: ApiClient, limit: int, user_id: Optional[int] = None, username: Optional[str] = None) → List[User]`

Retrieve the first x amount of followers from an account.

Either *user_id* or *username* need to be provided. If both are provided, the *user_id* takes precedence.

Parameters

- **client** – your ApiClient
- **user_id** – the user_id of the account to retrieve followers from
- **limit** – the maximum amount of followers to retrieve
- **username** – the username of the account to retrieve followers from

Returns

A list containing Instagram user objects (examples/objects/user.json).

`instauto.helpers.friendships.get_following(client: ApiClient, limit: int, user_id: Optional[int] = None, username: Optional[str] = None) → List[User]`

Retrieve the first x amount of users that an account is following.

Either *user_id* or *username* need to be provided. If both are provided, the *user_id* takes precedence.

Parameters

- **client** – your ApiClient
- **user_id** – the user_id of the account to retrieve following from
- **limit** – the maximum amount of users to retrieve
- **username** – the username of the account to retrieve following from

Returns

A list containing Instagram user objects (examples/objects/user.json).

`instauto.helpers.friendships.unfollow_user(client: ApiClient, user_id: Optional[int] = None, username: Optional[str] = None) → bool`

Unfollow a user.

Either *user_id* or *username* need to be provided. If both are provided, the *user_id* takes precedence.

Parameters

- **client** – your *ApiClient*
- **user_id** – the *user_id* of the account to unfollow
- **username** – the username of the account to unfollow

Returns

True if success else False

1.2.2 instauto.helpers.post module

`instauto.helpers.post.comment_post(client: ApiClient, media_id: str, comment: str) → bool`

Leave a comment on a post.

Parameters

- **client** – your *ApiClient*
- **media_id** – the post to comment on
- **comment** – the comment to place

Returns

True if success else False

`instauto.helpers.post.get_commenters_of_post(client: ApiClient, media_id: str) → List[User]`

Get users that commented on a post.

Parameters

- **client** – your *ApiClient*
- **media_id** – the post to retrieve the commenters from

Returns

A list of Instagram user objects (objects/post.json).

`instauto.helpers.post.get_likers_of_post(client: ApiClient, media_id: str) → List[User]`

Get users that liked a post.

Parameters

- **client** – your *ApiClient*
- **media_id** – the post to retrieve the likers from

Returns

A list of Instagram user objects (objects/user.json).

`instauto.helpers.post.like_post(client: ApiClient, media_id: str) → bool`

Like a post.

Parameters

- **client** – your *ApiClient*
- **media_id** – the post to like

Returns

True if success else False

`instauto.helpers.post.retrieve_posts_from_tag(client: ApiClient, tag: str, limit: int) → List[Post]`

Retrieve x amount of posts tagged with a tag.

Parameters

- **client** – your *ApiClient*
- **limit** – maximum amount of posts to retrieve
- **tag** – the tag to search for

Returns

A list of Instagram post objects (objects/post.json).

`instauto.helpers.post.retrieve_posts_from_user(client: ApiClient, limit: int, username: Optional[str] = None, user_id: Optional[int] = None) → List[Post]`

Retrieve x amount of posts from a user.

Either *user_id* or *username* need to be provided. If both are provided, the *user_id* takes precedence.

Parameters

- **client** – your *ApiClient*
- **limit** – maximum amount of posts to retrieve
- **username** – username of the account to retrieve posts from
- **user_id** – user_id of the account to retrieve posts from

Returns

A list of Instagram post objects (objects/post.json).

`instauto.helpers.post.retrieve_story_from_user(client: ApiClient, username: Optional[str] = None, user_id: Optional[int] = None) → List[Story]`

Retrieve x amount of posts from a user.

Either *user_id* or *username* need to be provided. If both are provided, the *user_id* takes precedence.

Parameters

- **client** – your *ApiClient*
- **limit** – maximum amount of posts to retrieve
- **username** – username of the account to retrieve posts from
- **user_id** – user_id of the account to retrieve posts from

Returns

A list of Instagram post objects (objects/post.json).

`instauto.helpers.post.save_post(client: ApiClient, media_id: str) → bool`

Save a post.

Parameters

- **client** – your *ApiClient*
- **media_id** – the media_id of a post

Returns

True if success else *False*

`instauto.helpers.post.unlike_post(client: ApiClient, media_id: str) → bool`

Undo the liking of a post.

Parameters

- **client** – your *ApiClient*
- **media_id** – the `media_id` of a post

Returns

True if success else *False*

`instauto.helpers.post.update_caption(client: ApiClient, media_id: str, new_caption: str) → bool`

Update the caption of a post.

Parameters

- **client** – your *ApiClient*
- **media_id** – the `media_id` of a post
- **new_caption** – the new caption

Returns

True if success else *False*

`instauto.helpers.post.upload_image_to_feed(client: ApiClient, image_path: str, caption: Optional[str] = None, location: Optional[Location] = None) → bool`

Upload an image to your feed. Location and caption are optional.

Parameters

- **client** – your *ApiClient*
- **image_path** – path to the image to upload
- **caption** – the caption of the post
- **location** – the location tag of the post

Returns

True if success else *False*

`instauto.helpers.post.upload_image_to_story(client: ApiClient, image_path: str) → bool`

Upload an image to your story.

Parameters

- **client** – your *ApiClient*
- **image_path** – path to the image to upload

Returns

True if success else *False*

1.2.3 instauto.helpers.search module

`instauto.helpers.search.get_user_by_username(client: ApiClient, username: str) → Optional[User]`

Retrieve a user by username.

Parameters

- **client** – your *ApiClient*
- **username** – username to search for

Returns

None if not found, else a user object (objects/user.json) containing the found user

`instauto.helpers.search.get_user_id_from_username(client: ApiClient, username: str) → Optional[int]`

Get the user id of a username.

Parameters

- **client** – your *ApiClient*
- **username** – username to search for

Returns

None if not found, else a user id of the found user

`instauto.helpers.search.search_tags(client: ApiClient, tag: str, limit: int) → List[dict]`

`instauto.helpers.search.search_username(client: ApiClient, username, count: int) → List[User]`

Search a username on Instagram.

Parameters

- **client** – your *ApiClient*
- **username** – username to search
- **count** – amount of results to retrieve

Returns

List of user objects (objects/user.json) that Instagram matched with the provider username

1.3 instauto.bot package

```
class instauto.bot.Bot(username: Optional[str] = None, password: Optional[str] = None, client:
    Optional[ApiClient] = None, delay_between_action: float = 2.0, delay_variance: float
    = 0.0)
```

```
    __init__(username: Optional[str] = None, password: Optional[str] = None, client: Optional[ApiClient] =
        None, delay_between_action: float = 2.0, delay_variance: float = 0.0) → None
```

Initiate a new *Bot* instance.

Parameters

- **username** – the username of the account
- **password** – the password of the account
- **client** – the *ApiClient* instance the Bot communicates with. If given, it will take precedence over credentials.

- **delay_between_action** – the amount of seconds to wait between actions (each like, follow, etc. is an action)
- **delay_variance** – the amount of variance to add to the delay. Delay will be random number between (delay - variance) - (delay + variance).

comment(*chance: int, amount: int, comments: List[str]*) → *Bot*

Comment on posts of users retrieved with the Input pipeline.

Parameters

- **chance** – integer between 0 and 100, represents a percentage between 0 and 100%. Defines the chance of this action being called for an account. Set to 25 to call on 1/4 of all accounts, 50 for 1/2 of all accounts, etc.
- **amount** – The amount of posts to comment on, if this action is being called for an account.
- **comments** – A random selected entry out of this list will be used as text to comment.

follow(*chance: int*) → *Bot*

Follow users retrieved with the Input pipeline.

Parameters

chance – integer between 0 and 100, represents a percentage between 0 and 100%. Defines the chance of this action being called for an account. Set to 25 to call on 1/4 of all accounts, 50 for 1/2 of all accounts, etc.

classmethod from_client(*client: ApiClient, delay_between_action: float = 2.0, delay_variance: float = 0.0*) → *Bot*

like(*chance: int, amount: int*) → *Bot*

Like posts of users retrieved with the Input pipeline.

Parameters

- **chance** – integer between 0 and 100, represents a percentage between 0 and 100%. Defines the chance of this action being called for an account. Set to 25 to call on 1/4 of all accounts, 50 for 1/2 of all accounts, etc.
- **amount** – The amount of posts to like, if this action is being called for an account.

start()

Start the bot.

Once the bot is started, it will run until it went through all retrieved accounts, or if the *stop* attribute is set to *True*.

class instauto.bot.Input(*client: ApiClient*)

from_commenters_of(*account_name: str, limit: int*) → *Input*

Retrieves accounts that have commented on recent posts of *account_name*.

Parameters

- **account_name** – the account to retrieve from
- **limit** – the amount of accounts to retrieve

from_followers_of(*account_name: str, limit: int*) → *Input*

Retrieves accounts that follow *account_name*.

Parameters

- **account_name** – the account to retrieve from

- **limit** – the amount of accounts to retrieve

from_following_of(*account_name: str, limit: int*) → *Input*

Retrieves accounts that *account_name* follows.

Parameters

- **account_name** – the account to retrieve from
- **limit** – the amount of accounts to retrieve

from_likers_of(*account_name: str, limit: int*) → *Input*

Retrieves accounts that have liked recent posts of *account_name*.

Parameters

- **account_name** – the account to retrieve from
- **limit** – the amount of accounts to retrieve

from_user_list(*accounts: List[User]*) → *Input*

Add supplied accounts to input

Parameters

accounts – List of account objects (objects/user.json)

get_posts(*account_name: str, force: bool = False*) → List[Post]

PYTHON MODULE INDEX

i

- `instauto.api.actions.authentication`, 3
- `instauto.api.actions.challenge`, 4
- `instauto.api.actions.direct`, 4
- `instauto.api.actions.friendships`, 5
- `instauto.api.actions.helpers`, 5
- `instauto.api.actions.post`, 6
- `instauto.api.actions.profile`, 7
- `instauto.api.actions.request`, 7
- `instauto.api.actions.search`, 7
- `instauto.api.actions.stub`, 8
- `instauto.api.client`, 8
- `instauto.api.constants`, 8
- `instauto.api.exceptions`, 9
- `instauto.api.structs`, 9
- `instauto.helpers.friendships`, 16
- `instauto.helpers.post`, 17
- `instauto.helpers.search`, 20

Symbols

`__init__()` (*instauto.bot.Bot* method), 20

A

`admin_user_ids` (*instauto.api.structs.Thread* attribute), 15

`android_id` (*instauto.api.structs.State* property), 14

`android_release` (*instauto.api.structs.DeviceProfile* attribute), 10

`android_sdk_version` (*instauto.api.structs.DeviceProfile* attribute), 10

`ApiClient` (class in *instauto.api.client*), 8

`AuthenticationMixin` (class in *instauto.api.actions.authentication*), 3

`authorization` (*instauto.api.structs.State* attribute), 14

`AuthorizationError`, 9

B

`BadResponse`, 9

`bandwidth_speed_kbps` (*instauto.api.structs.State* property), 14

`bc_hmac` (*instauto.api.client.ApiClient* attribute), 8

`blended_inbox_enabled` (*instauto.api.structs.Inbox* attribute), 11

`Bot` (class in *instauto.bot*), 20

`breadcrumb_private_key` (*instauto.api.client.ApiClient* attribute), 8

`build_number` (*instauto.api.structs.IGProfile* attribute), 11

C

`capabilities` (*instauto.api.structs.IGProfile* attribute), 11

`ChallengeChoice` (class in *instauto.api.structs*), 9

`ChallengeMixin` (class in *instauto.api.actions.challenge*), 4

`change_password()` (*instauto.api.actions.authentication.AuthenticationMixin* method), 3

`chipset` (*instauto.api.structs.DeviceProfile* attribute), 10

`comment()` (*instauto.bot.Bot* method), 21

`comment_post()` (in module *instauto.helpers.post*), 17

`connection_speed` (*instauto.api.structs.State* property), 14

`CorruptedSaveData`, 9

D

`DEFAULT_SIGNATURE_KEY` (in module *instauto.api.constants*), 8

`device` (*instauto.api.structs.DeviceProfile* attribute), 10

`device_profile` (*instauto.api.actions.authentication.AuthenticationMixin* attribute), 3

`device_profile` (*instauto.api.actions.challenge.ChallengeMixin* attribute), 4

`device_profile` (*instauto.api.actions.direct.DirectMixin* attribute), 4

`device_profile` (*instauto.api.actions.friendships.FriendshipsMixin* attribute), 5

`device_profile` (*instauto.api.actions.helpers.HelperMixin* attribute), 5

`device_profile` (*instauto.api.actions.post.PostMixin* attribute), 6

`device_profile` (*instauto.api.actions.profile.ProfileMixin* attribute), 7

`device_profile` (*instauto.api.actions.request.RequestMixin* attribute), 7

`device_profile` (*instauto.api.actions.search.SearchMixin* attribute), 7

`device_profile` (*instauto.api.actions.stub.StubMixin* attribute), 8

`device_profile` (*instauto.api.client.ApiClient* attribute), 8

`DeviceProfile` (class in *instauto.api.structs*), 9

`direct_get_thread()` (*instauto.api.actions.direct.DirectMixin* method), 4

`direct_send()` (*instauto.api.actions.direct.DirectMixin* method), 4

`direct_update_inbox()` (*instauto.api.actions.direct.DirectMixin* method), 4

`DirectMixin` (class in *instauto.api.actions.direct*), 4

`dpi` (*instauto.api.structs.DeviceProfile* attribute), 10

E

email (*instauto.api.structs.ChallengeChoice* attribute), 9

F

Feed (*instauto.api.structs.PostLocation* attribute), 13

female (*instauto.api.structs.WhichGender* attribute), 15

fill() (*instauto.api.structs.State* method), 14

follow() (*instauto.bot.Bot* method), 21

follow_list (*instauto.api.structs.Surface* attribute), 15

follow_request_approve() (*instauto.api.actions.friendships.FriendshipsMixin* method), 5

follow_requests (*instauto.api.structs.Surface* attribute), 15

follow_requests_get() (*instauto.api.actions.friendships.FriendshipsMixin* method), 5

follow_user() (*instauto.helpers.friendships*), 16

follower_remove() (*instauto.api.actions.friendships.FriendshipsMixin* method), 5

follower_show() (*instauto.api.actions.friendships.FriendshipsMixin* method), 5

followers_get() (*instauto.api.actions.friendships.FriendshipsMixin* method), 5

following_get() (*instauto.api.actions.friendships.FriendshipsMixin* method), 5

following_list (*instauto.api.structs.Surface* attribute), 15

FriendshipsMixin (*instauto.api.actions.friendships*), 5

from_client() (*instauto.bot.Bot* class method), 21

from_commenters_of() (*instauto.bot.Input* method), 21

from_followers_of() (*instauto.bot.Input* method), 21

from_following_of() (*instauto.bot.Input* method), 22

from_json() (*instauto.api.client.ApiClient* class method), 8

from_likers_of() (*instauto.bot.Input* method), 22

from_user_list() (*instauto.bot.Input* method), 22

G

GET (*instauto.api.structs.Method* attribute), 13

get_commenters_of_post() (*instauto.helpers.post*), 17

get_followers() (*instauto.helpers.friendships*), 16

get_following() (*instauto.helpers.friendships*), 16

get_image_type() (*instauto.api.actions.helpers.HelperMixin* static method), 5

get_likers_of_post() (*instauto.helpers.post*), 17

get_posts() (*instauto.bot.Input* method), 22

get_user_by_username() (*instauto.helpers.search*), 20

get_user_id_from_username() (*instauto.helpers.search*), 20

H

has_older (*instauto.api.structs.Inbox* attribute), 12

has_pending_top_requests (*instauto.api.structs.Inbox* attribute), 12

HelperMixin (*instauto.api.actions.helpers*), 5

http_engine (*instauto.api.structs.IGProfile* attribute), 11

I

id (*instauto.api.structs.IGProfile* attribute), 11

ig_profile (*instauto.api.actions.authentication.AuthenticationMixin* attribute), 3

ig_profile (*instauto.api.actions.challenge.ChallengeMixin* attribute), 4

ig_profile (*instauto.api.actions.direct.DirectMixin* attribute), 4

ig_profile (*instauto.api.actions.friendships.FriendshipsMixin* attribute), 5

ig_profile (*instauto.api.actions.helpers.HelperMixin* attribute), 5

ig_profile (*instauto.api.actions.post.PostMixin* attribute), 6

ig_profile (*instauto.api.actions.profile.ProfileMixin* attribute), 7

ig_profile (*instauto.api.actions.request.RequestMixin* attribute), 7

ig_profile (*instauto.api.actions.search.SearchMixin* attribute), 7

ig_profile (*instauto.api.actions.stub.StubMixin* attribute), 8

ig_profile (*instauto.api.client.ApiClient* attribute), 8

IGProfile (*instauto.api.structs*), 10

Inbox (*instauto.api.structs*), 11

inbox (*instauto.api.actions.direct.DirectMixin* property), 4

IncorrectLoginDetails, 9

initiate_from_file() (*instauto.api.client.ApiClient* class method), 8

Input (*instauto.bot*), 21

instauto.api.actions.authentication module, 3

instauto.api.actions.challenge module, 4

instauto.api.actions.direct
 module, 4
 instauto.api.actions.friendships
 module, 5
 instauto.api.actions.helpers
 module, 5
 instauto.api.actions.post
 module, 6
 instauto.api.actions.profile
 module, 7
 instauto.api.actions.request
 module, 7
 instauto.api.actions.search
 module, 7
 instauto.api.actions.stub
 module, 8
 instauto.api.client
 module, 8
 instauto.api.constants
 module, 8
 instauto.api.exceptions
 module, 9
 instauto.api.structs
 module, 9
 instauto.helpers.friendships
 module, 16
 instauto.helpers.post
 module, 17
 instauto.helpers.search
 module, 20
 InvalidUserId, 9
 items (*instauto.api.structs.Thread* attribute), 15

L

left_users (*instauto.api.structs.Thread* attribute), 15
 like() (*instauto.bot.Bot* method), 21
 like_post() (in module *instauto.helpers.post*), 17
 log_in() (*instauto.api.actions.authentication.AuthenticationMixin* method), 3
 logged_in_account_data (*instauto.api.structs.State* attribute), 14
 LoggedInAccountData (class in *instauto.api.structs*), 12

M

male (*instauto.api.structs.WhichGender* attribute), 15
 manufacturer (*instauto.api.structs.DeviceProfile* attribute), 10
 Method (class in *instauto.api.structs*), 13
 mid (*instauto.api.structs.State* attribute), 14
 MissingValue, 9
 model (*instauto.api.structs.DeviceProfile* attribute), 10
 module
 instauto.api.actions.authentication, 3

instauto.api.actions.challenge, 4
 instauto.api.actions.direct, 4
 instauto.api.actions.friendships, 5
 instauto.api.actions.helpers, 5
 instauto.api.actions.post, 6
 instauto.api.actions.profile, 7
 instauto.api.actions.request, 7
 instauto.api.actions.search, 7
 instauto.api.actions.stub, 8
 instauto.api.client, 8
 instauto.api.constants, 8
 instauto.api.exceptions, 9
 instauto.api.structs, 9
 instauto.helpers.friendships, 16
 instauto.helpers.post, 17
 instauto.helpers.search, 20

N

next_cursor (*instauto.api.structs.Inbox* attribute), 12
 NoAuthDetailsProvided, 9
 NotFoundError, 9
 Null (*instauto.api.structs.PostLocation* attribute), 13

O

oldest_cursor (*instauto.api.structs.Inbox* attribute), 12
 other (*instauto.api.structs.WhichGender* attribute), 15

P

pending_requests_total (*instauto.api.structs.Inbox* attribute), 12
 phone (*instauto.api.structs.ChallengeChoice* attribute), 9
 POST (*instauto.api.structs.Method* attribute), 13
 post_archive() (*instauto.api.actions.post.PostMixin* method), 6
 post_carousel() (*instauto.api.actions.post.PostMixin* method), 6
 post_comment() (*instauto.api.actions.post.PostMixin* method), 6
 post_get_commenters() (*instauto.api.actions.post.PostMixin* method), 6
 post_get_comments() (*instauto.api.actions.post.PostMixin* method), 6
 post_get_likers() (*instauto.api.actions.post.PostMixin* method), 6
 post_like() (*instauto.api.actions.post.PostMixin* method), 6
 post_post() (*instauto.api.actions.post.PostMixin* method), 6
 post_retrieve_by_id() (*instauto.api.actions.post.PostMixin* method), 6

post_retrieve_by_tag() (in-
stauto.api.actions.post.PostMixin method),
6
post_retrieve_by_user() (in-
stauto.api.actions.post.PostMixin method),
6
post_retrieve_story() (in-
stauto.api.actions.post.PostMixin method),
6
post_save() (*instauto.api.actions.post.PostMixin*
method), 6
post_unarchive() (*instauto.api.actions.post.PostMixin*
method), 6
post_unlike() (*instauto.api.actions.post.PostMixin*
method), 7
post_update_caption() (in-
stauto.api.actions.post.PostMixin method),
7
PostLocation (class in *instauto.api.structs*), 13
PostMixin (class in *instauto.api.actions.post*), 6
prefer_not_to_say (*instauto.api.structs.WhichGender*
attribute), 15
prev_cursor (*instauto.api.structs.Inbox* attribute), 12
profile (*instauto.api.structs.Surface* attribute), 15
profile_info() (*instauto.api.actions.profile.ProfileMixin*
method), 7
profile_set_biography() (in-
stauto.api.actions.profile.ProfileMixin method),
7
profile_set_gender() (in-
stauto.api.actions.profile.ProfileMixin method),
7
profile_set_picture() (in-
stauto.api.actions.profile.ProfileMixin method),
7
profile_update() (in-
stauto.api.actions.profile.ProfileMixin method),
7
ProfileMixin (class in *instauto.api.actions.profile*), 7
properties (*instauto.api.structs.Thread* attribute), 15

R

refresh() (*instauto.api.structs.State* method), 15
RequestMixin (class in *instauto.api.actions.request*), 7
resolution (*instauto.api.structs.DeviceProfile* at-
tribute), 10
retrieve_posts_from_tag() (in module in-
stauto.helpers.post), 17
retrieve_posts_from_user() (in module in-
stauto.helpers.post), 18
retrieve_story_from_user() (in module in-
stauto.helpers.post), 18

S

save_post() (in module *instauto.helpers.post*), 18
save_to_disk() (*instauto.api.client.ApiClient* method),
8
search_tag() (*instauto.api.actions.search.SearchMixin*
method), 7
search_tags() (in module *instauto.helpers.search*), 20
search_username() (in module in-
stauto.helpers.search), 20
search_username() (in-
stauto.api.actions.search.SearchMixin method),
7
SearchMixin (class in *instauto.api.actions.search*), 7
seq_id (*instauto.api.structs.Inbox* attribute), 12
signature_key (*instauto.api.structs.IGProfile* at-
tribute), 11
snapshot_at_ms (*instauto.api.structs.Inbox* attribute),
12
start() (*instauto.bot.Bot* method), 21
startup_country (*instauto.api.structs.State* property),
15
State (class in *instauto.api.structs*), 13
state (*instauto.api.actions.authentication.AuthenticationMixin*
attribute), 3
state (*instauto.api.actions.challenge.ChallengeMixin* at-
tribute), 4
state (*instauto.api.actions.direct.DirectMixin* attribute),
4
state (*instauto.api.actions.friendships.FriendshipsMixin*
attribute), 5
state (*instauto.api.actions.helpers.HelperMixin* at-
tribute), 6
state (*instauto.api.actions.post.PostMixin* attribute), 7
state (*instauto.api.actions.profile.ProfileMixin* at-
tribute), 7
state (*instauto.api.actions.request.RequestMixin* at-
tribute), 7
state (*instauto.api.actions.search.SearchMixin* at-
tribute), 8
state (*instauto.api.actions.stub.StubMixin* attribute), 8
state (*instauto.api.client.ApiClient* attribute), 8
StateExpired, 9
Story (*instauto.api.structs.PostLocation* attribute), 13
StubMixin (class in *instauto.api.actions.stub*), 8
Surface (class in *instauto.api.structs*), 15

T

Thread (class in *instauto.api.structs*), 15
thread_id (*instauto.api.structs.Thread* attribute), 15
thread_v2_id (*instauto.api.structs.Thread* attribute), 15
threads (*instauto.api.structs.Inbox* attribute), 12
to_json() (*instauto.api.client.ApiClient* method), 8

U

[unfollow_user\(\)](#) (in module *instauto.helpers.friendships*), [16](#)
[unlike_post\(\)](#) (in module *instauto.helpers.post*), [18](#)
[unseen_count](#) (*instauto.api.structs.Inbox* attribute), [12](#)
[unseen_count_ts](#) (*instauto.api.structs.Inbox* attribute), [12](#)
[update_caption\(\)](#) (in module *instauto.helpers.post*), [19](#)
[upload_image_to_feed\(\)](#) (in module *instauto.helpers.post*), [19](#)
[upload_image_to_story\(\)](#) (in module *instauto.helpers.post*), [19](#)
[user_follow\(\)](#) (*instauto.api.actions.friendships.FriendshipsMixin* method), [5](#)
[user_unfollow\(\)](#) (*instauto.api.actions.friendships.FriendshipsMixin* method), [5](#)
[users](#) (*instauto.api.structs.Thread* attribute), [15](#)

V

[valid](#) (*instauto.api.structs.State* property), [15](#)
[version](#) (*instauto.api.structs.IGProfile* attribute), [11](#)

W

[WhichGender](#) (class in *instauto.api.structs*), [15](#)
[www_claim](#) (*instauto.api.structs.State* attribute), [14](#)