



# INSTAPY

PYTHONIC TOOL FOR AUTOMATED INSTAGRAM INTERACTIONS

by Tim Grossmann

# PROJECT SELECTION



1

---

It is a quite **intriguing** project that piques my **interest**.

2

---

Its structure, code & documentation are **easy to comprehend**.

3

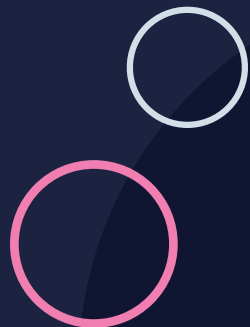
---

It is continuously **developing and** **receptive** to PRs.



# PROJECT COMPREHENDING

- Read the entire **code & documentation** in order to understand the project deeper & analyze the need for new features.
- **Hands-on practice** on using the software as a simple user.
- Started focusing into **my own feature ideas** of enhancing the project.
- Checked with **github issues section** for asked features.

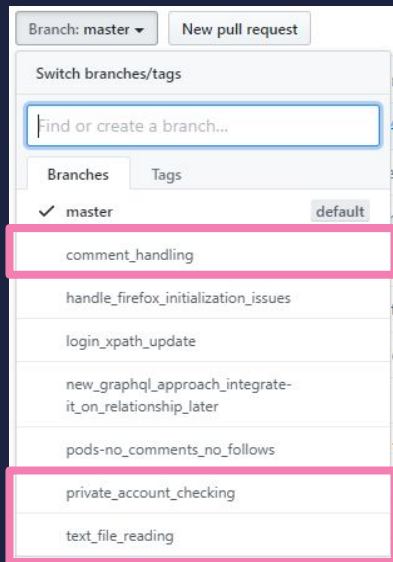


# COMMUNICATION WITH TEAM

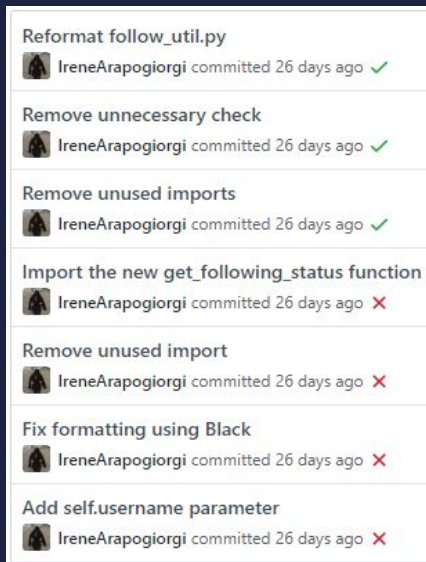
- **Channels:** GitHub (issues & PRs), Discord
- **Reviewers:** Tim Grossmann (creator), Felix Breuer (contributor)
- **Received:** small tips, answers to questions, continuous feedback & code review

# SOURCE CODE MANAGEMENT

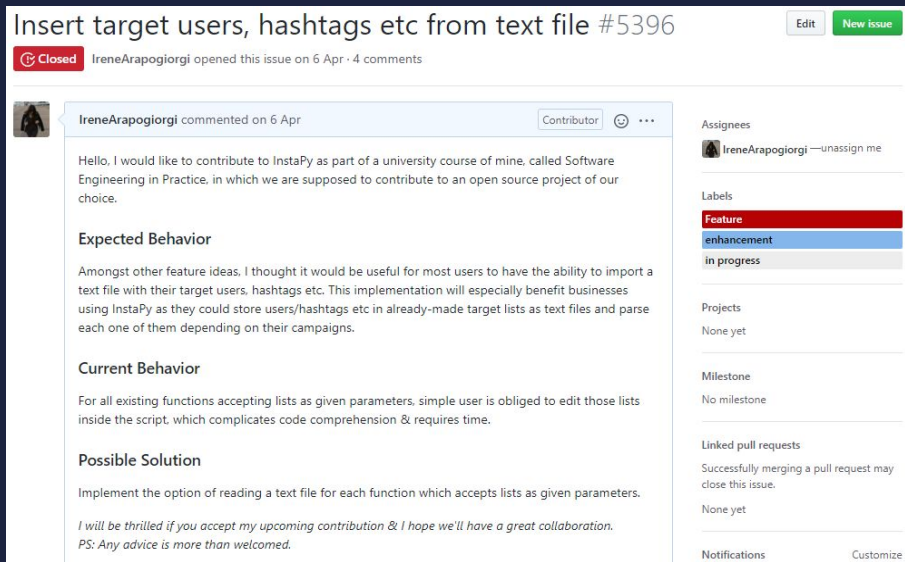
create new branches



make commits



open issues



# FIRST CONTRIBUTION

Insert target users,  
hashtags etc from text files.

# WHY THOUGH?

1

---

There is **need**  
for automation.

2

---

Multiple functions  
will be **affected**.

3

---

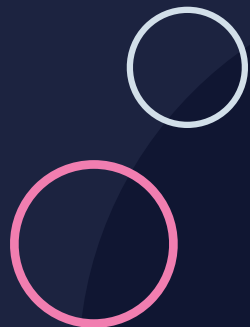
**Implementation**  
was clear in mind.

# IMPLEMENTATION OVERVIEW

7 files changed

94 additions, 0 deletions

2 new functions, unit testing, documentation, changelog





# IMPLEMENTATION

## quickstart.py

```
# Like posts based on hashtags
hashtags = session.target_list("C:\\Users\\.....\\hashtags.txt")
session.like_by_tags(hashtags, amount=10)

# Follow the followers of each given user
users = session.target_list("C:\\Users\\.....\\users.txt")
session.follow_user_followers(users, amount=10, randomize=False)
```

## instapy.py

```
def target_list(self, file):
    """ Extracts target list from text file """
    target_list = file_handling(file)

    if "FileNotFoundError" in target_list:
        self.logger.warning("No such file or directory: " + file + "\n")
        return []

    return target_list
```

## util.py

```
2403 def file_handling(file):
2404     """ Extracts text file's elements """
2405     elements = []
2406     try:
2407         with open(file, "r") as f:
2408             # extract file's lines in list
2409             for line in f.readlines():
2410                 if line != "\n":
2411                     # remove leading whitespaces, newline and tab characters
2412                     element = line.lstrip().strip("\n")
2413                     elements.append(element)
2414     except FileNotFoundError:
2415         return ["FileNotFoundError"]
2416
2417     return elements
```

# FUNCTIONALITY CHECK

```
.....  
INFO [2020-05-16 21:09:24] [irene_arp] Logged in successfully!  
.....  
INFO [2020-05-16 21:09:24] [irene_arp] Saving account progress...  
INFO [2020-05-16 21:09:40] [irene_arp] Tag [1/2]  
INFO [2020-05-16 21:09:40] [irene_arp] -> b'photography'  
INFO [2020-05-16 21:09:50] [irene_arp] desired amount: 10 | top posts [disabled]: 9 | possible posts: 567827651  
INFO [2020-05-16 21:09:57] [irene_arp] Like# [1/10]  
INFO [2020-05-16 21:09:57] [irene_arp] https://www.instagram.com/p/CAQh6bpH_7C/  
INFO [2020-05-16 21:10:01] [irene_arp] Image from: b'marielopareva'  
INFO [2020-05-16 21:10:01] [irene_arp] Link: b'https://www.instagram.com/p/CAQh6bpH_7C/'
```

Successful use

```
.....  
WARNING [2020-05-16 21:17:50] [irene_arp] No such file or directory: C:\Users\irene\Desktop\hashtags.txt
```

FileNotFoundError

# TESTING

unit testing using `unittest`

code coverage using `coverage.py`

3 test cases:

- valid entries
- invalid entries
- invalid file path

continuous integration with `Travis CI`

Coverage for `test_file_handling.py` : 100%

13 statements

13 run

0 missing

0 excluded

```
1 import unittest
2
3 from instapy import util
4
5
6 class TestFileHandling(unittest.TestCase):
7     """Unit tests for file_handling function"""
8
9     # List of expected entries
10    global expected_entries
11    expected_entries = ["photography", "portrait", "minimalistic"]
12
13    def test_valid_entries(self):
14        """Check file_handling with valid entries"""
15        actual_entries = util.file_handling("resources/valid_target_list.txt")
16        self.assertEqual(actual_entries, expected_entries)
17
18
19    def test_invalid_entries(self):
20        """Check file_handling with invalid entries"""
21        actual_entries = util.file_handling("resources/invalid_target_list.txt")
22        self.assertEqual(actual_entries, expected_entries)
23
24
25    def test_invalid_filepath(self):
26        """Check whether it raises an exception when the filepath is invalid"""
27        actual_entries = util.file_handling("invalid_filepath")
28        self.assertEqual(actual_entries, ["FileNotFoundError"])
```

# CODE QUALITY

```
irene@LAPTOP-SL23CI1H MINGW64 ~/GitRepositories/InstaPy (text_file_reading)
$ black instapy/util.py
All done! 0 0 00 00
1 file reformatted.

irene@LAPTOP-SL23CI1H MINGW64 ~/GitRepositories/InstaPy (text_file_reading)
$ black instapy/instapy.py
reformatted instapy\instapy.py
All done! 0 0 00 00
1 file reformatted.

irene@LAPTOP-SL23CI1H MINGW64 ~/GitRepositories/InstaPy (text_file_reading)
$ black tests/test_file_handling.py
reformatted tests\test_file_handling.py
All done! 0 0 00 00
1 file reformatted.
```

- code documentation
- code formatting using Black
- linting using pylint

# PULL REQUEST

## - CODE REVIEW

## Insert target users, hashtags etc from text file - Issue #5396 #5443

**Merged** breuerfelix merged 18 commits into `tingrossmann:master` from `IreneArapogiorgi:text_file_reading` 13 days ago

Conversation 4 Commits 18 Checks 0 Files changed 7 +94 -0



IreneArapogiorgi commented 16 days ago • edited

Contributor

Hello, I was assigned to implement the ability to parse text files containing target users, hashtags, comments etc in order to use it with functions which accept lists as given parameters. This will be a part of a university assignment of mine as I also mentioned on issue #5396.

**Functions added:**  
file\_handling in util.py  
target\_list in instapy.py

*PS: At first, I implemented file\_handling function inside instapy.py (with only a few changes) & that also worked, but I had to change that implementation because I wasn't able to do some unit testing.*

**Unit testing:**  
test\_file\_handling.py in tests folder  
invalid\_target\_list.txt in tests/resources folder  
valid\_target\_list.txt in tests/resources folder

**Example of using new feature in quickstart.py**

```
""" Quickstart script for InstaPy usage """

# imports
from instapy import InstaPy
from instapy import smart_run
from instapy import set_workspace

# set workspace folder at desired location (default is at your home folder)
set_workspace(path=None)

# login credentials
insta_username = "your instagram username"
insta_password = "your instagram password"

# get an InstaPy session
session = InstaPy(username=insta_username,
                  password=insta_password,
                  headless_browser=False)

with smart_run(session):

    # like posts based on hashtags
    hashtags = session.target_list("C:\\Users\\.....\\hashtags.txt")
    session.like_by_tags(hashtags, amount=9)
```

If there are any changes/improvements needed, please let me know so that I can work on them.

Edit

Reviewers

breuerfelix ✓

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Unsubscribe

You're receiving notifications because you authored the thread.

2 participants



Allow edits by maintainers

breuerfelix commented 9 days ago

Collaborator

Clean code, looking really good!

Would you mind also adding a quick line in the changelog file and also put some example in the documentation ? somewhere in the section about like by tags :) )

otherwise nobody will ever know that this function exists cause the majority is not looking in the code and only have a look at the documentation :)

Branch: master

Commits on May 1, 2020

Insert target users, hashtags etc from text file - Issue #5396 (#5443)

IreneArapogiorgi committed 13 days ago

Verified



8f1a129



# DOCUMENTATION

## Target Lists

This is used to parse text files containing target lists of users, hashtags, comments etc

For example:

```
# Like posts based on hashtags
hashtags = session.target_list("C:\\Users\\.....\\hashtags.txt")
session.like_by_tags(hashtags, amount=10)

# Follow the followers of each given user
users = session.target_list("C:\\Users\\.....\\users.txt")
session.follow_user_followers(users, amount=10, randomize=False)
```

Note that your text file should look like this:

```
hashtag1
hashtag2
hashtag3
```

or

```
user1
user2
user3
```

Functions you can use `target_list` with:

```
story_by_user , story_by_tag , like_by_tags , follow_by_tags , follow_user_followers , follow_user_following ,
follow_likers , follow_commenters , follow_by_list , set_skip_users , set_ignore_users , set_dont_include ,
interact_by_users , interact_by_users_tagged_posts , interact_user_followers , interact_user_following ,
interact_by_comments , set_comments , set_comment_replies , set_mandatory_words , unfollow_users
```

## SECOND CONTRIBUTION

Check if user's account  
is not private.



# WHY THOUGH?

```
49     # Get followers count
50     followers_count, _ = get_relationship_counts(browser, username, logger)
51
52     if grab != "full" and grab > followers_count:
53         logger.info(
54             "You have requested higher amount than existing followers count "
55             " ~gonna grab all available"
56         )
57         grab = followers_count
58
59     # TO-DO: Check if user's account is not private
60
```

To-do comments inside get\_followers & get\_following functions in file relationship\_tools.py

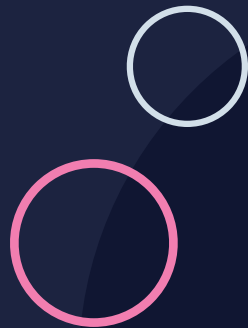


# IMPLEMENTATION OVERVIEW

5 files changed

200 additions, 116 deletions

12 functions modified, 1 new file added



# IMPLEMENTATION

```
62     # Check if user's account is private and we don't follow
63     following_status, _ = get_following_status(
64         browser, "profile", self_username, username, None, logger, logfolder
65     )
66
67     is_private = is_private_profile(browser, logger, following_status == "Following")
68     if (
69         is_private is None
70         or (is_private is True and following_status not in ["Following", True])
71         or (following_status == "Blocked")
72     ):
73         logger.info("This user is private and we are not following")
74         return False
```

get\_followers() & get\_following() in relationship\_tools.py

# IMPLEMENTATION

get\_following\_status()

from unfollow\_util.py file  
to follow\_util.py file

```
110 instapy/follow_util.py
...  ...  @@ -0,0 +1,110 @@
1  + """ Module which handles the follow features """
2  +
3  + from .util import web_address_navigator
4  + from .util import get_username_from_id
5  + from .util import is_page_available
6  + from .util import update_activity
7  + from .util import explicit_wait
8  + from .util import load_user_id
9  +
10 + from selenium.common.exceptions import NoSuchElementException
11 +
12 + from .xpath import read_xpath
13 +
14 +
15 + def get_following_status(
16 +     browser, track, username, person, person_id, logger, logfolder
17 + ):
18 +     """ Verify if you are following the user in the loaded page """
```

# IMPLEMENTATION

```
▼ 6 instapy/instapy.py
@@ -4309,6 +4309,7 @@ def grab_followers(
    4309 4309      # Get `followers` data
    4310 4310      grabbed_followers = get_followers(
    4311 4311          self.browser,
    4312 +      self.username,
    4312 4313          username,
    4313 4314          amount,
    4314 4315          self.relationship_data,
```

Parse `self.username` parameter directly

Fix related imports

```
▼ 2 instapy/like_util.py
@@ -17,7 +17,7 @@
    17 17      from .util import explicit_wait
    18 18      from .util import extract_text_from_element
    19 19      from .quota_supervisor import quota_supervisor
    20 - from .unfollow_util import get_following_status
    20 + from .follow_util import get_following_status
    21 21      from .event import Event
```

# IMPLEMENTATION

```
@@ -694,6 +736,7 @@ def get_nonfollowers(  
694 736     # get `Followers` data  
695 737     all_followers = get_followers(  
696 738         browser,  
739 +         self_username,  
697 740         username,  
698 741         "full",  
699 742         relationship_data,  
@@ -702,9 +745,15 @@ def get_nonfollowers(  
702 745         logger,  
703 746         logfolder,  
704 747     )  
748 +  
749 +     # if all_followers is False we have targeted a private account that we don't follow  
750 +     if not all_followers:  
751 +         return False
```

Parse self.username parameter through self\_username & check if need to break

# FUNCTIONALITY CHECK

```
.....
INFO [2020-05-18 00:31:44] [irene_ar] Logged in successfully!
.....
INFO [2020-05-18 00:31:44] [irene_ar] Saving account progress...

INFO [2020-05-18 00:31:58] [irene_ar] Starting to get the `Followers` data..
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
INFO [2020-05-18 00:31:58] [irene_ar] Retrieving live `Followers` data of ██████████ at the range of 10
48/10 [=====+] 480% |> Elapsed: 0.03 seconds |> ETA: 0.02 seconds

INFO [2020-05-18 00:32:29] [irene_ar] Grabbed 48 usernames from `Followers` as requested at 0.03 seconds
INFO [2020-05-18 00:32:29] [irene_ar] Stored `Followers` data at C:\Users\irene\InstaPy\logs\irene_ar\relationship_data/██████████ /followers/18-05-2020~10~48.json local file
```


Grab followers of a private account we follow

```
INFO [2020-05-18 00:36:00] [irene_ar] Starting to get the `Followers` data..
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
INFO [2020-05-18 00:36:00] [irene_ar] Retrieving live `Followers` data of ██████████ at the range of 10
INFO [2020-05-18 00:36:04] [irene_ar] You have requested higher amount than existing followers count ~gonna grab all available
INFO [2020-05-18 00:36:05] [irene_ar] This user is private and we are not following
```

Grab followers of a private account we don't follow

# PULL REQUEST

## - CODE REVIEW




**breuerfelix** commented 23 days ago Collaborator

thanks for this PR :)

I hope to find the time to look into this and test it out.  
I just got one question right of the start.  
What does happen when you pass 'unneeded' as a parameter ?  
a cleaner solution would be to append `self_username` as the last argument of this function and set the default value to `None`. so all the current functions using it do not have to get touched since there is a default value for that :)

also `None` is way clearer than a string with 'unneeded'.


I am still curious because i do not see any check if the variable is "unneeded" or not. So the Bot will try to interpret unneeded as a username right ?



**IreneArapogiorgi** commented 22 days ago • edited Author Contributor

@breuerfelix I changed the implementation replacing the "unneeded" string with the actual `self_username` for all relevant functions, as we need to check whether the target user is private and whether we follow him inside all these functions before we proceed.

Let me know if there is anything else I need to improve/change.



**breuerfelix** commented yesterday Collaborator

perfect ! looks way better now :) and actually just by looking at the PR there shouldnt be any errors :)

now we just need to get someone individual to test this PR :D


are you able to find someone (maybe in discord or a friend or smth like that) who can test out the old and new functions with a simple script ? i will also try to do some tests but with a more complex PR more testers are always better :)

## Check if user's account is private - Issue #5451 #5470

Edit Open with

Open IreneArapogiorgi wants to merge 19 commits into `tingrossmann:master` from `IreneArapogiorgi:private_account_checking`

Conversation 4 Commits 19 Checks 0 Files changed 6 +200 -116



**IreneArapogiorgi** commented 26 days ago • edited Contributor

Hello, I was assigned to implement checking whether target user's account is private or not in order to proceed into finding his followers/followings. This will be a part of a university assignment of mine as I also mentioned on issue #5451.

About my implementation, I used `is_private_profile()` to check if target user's account is private or not. In addition to that, I created a new file `follow_util.py` in which I moved the implementation of `get_following_status()` and then I imported it inside both `unfollow_util.py` and `like_util.py`, where it was already needed and used, and then inside `relationship_tools.py` in order to check whether the user is following the targeted private account, so in this case we can proceed into finding his followers and followings.

**Files added:**  
`follow_util.py`

**Files changed:**  
`unfollow_util.py`  
`relationship_tools.py`  
`like_util.py`  
`instapy.py`  
`util.py`

If there are any changes/improvements needed, please let me know so that I can work on them. Also, any advice about unit testing my implementation is welcomed.

Reviewers

No reviews

Still in progress? Convert to draft

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Successfully merging this pull request may close these issues.

None yet

# TESTING

□ unit testing required parsing personal data



testing out all possible cases for affected functions



breuerfelix commented yesterday

Collaborator



perfect ! looks way better now :) and actually just by looking at the PR there shouldnt be any errors :)

now we just need to get someone individual to test this PR :D

are you able to find someone (maybe in discord or a friend or smth like that) who can test out the old and new functions with a simple script ? i will also try to do some tests but with a more complex PR more testers are always better :)

IreneArapogiorgi commented yesterday

Author

Contributor



Absolutely! I also made a script which might help with testing all possible cases. For each of the relative functions, you can run seperately the 3 possible senarios - we cannot check all cases by running the script once, obviously. For me everything works fine.

```
""" Quickstart script for InstaPy usage """
```





**THANK YOU**

*Serene*

*To be continued...*