
Collecting User Info

User Profiles, Friends & Followers

Outline

- User Data in Tweet record
 - HCDE user module UserLookup.py
 - HCDE user module FriendsFollowers.py
-

Tweet JSON Structure

- When requesting tweets the JSON structure includes some user data
 - The “user” structure within the tweet data
 - Consider the `sample_tweets_json.txt` file
-

Almost everything

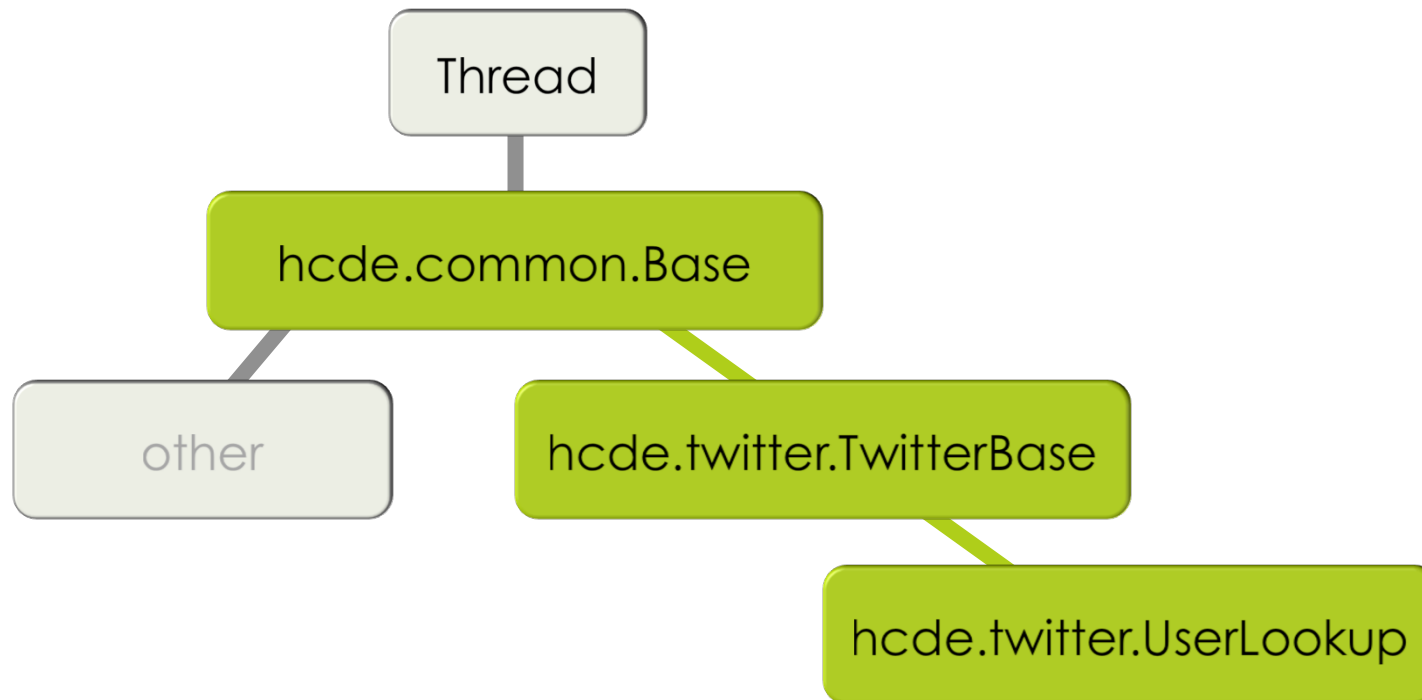
```
"user": {
  "contributors_enabled": false,
  "created_at": "Sat Apr 09 02:08:26 +0000 2011",
  "default_profile": true,
  "default_profile_image": false,
  "description": "Just a young White boy trying to live a life like Young Jeezy.",
  "entities": {
    "description": {
      "urls": []
    }
  },
  "favourites_count": 1097,
  "follow_request_sent": false,
  "followers_count": 310,
  "following": false,
  "friends_count": 400,
  "geo_enabled": false,
  "id": 279330988,
  "id_str": "279330988",
  "is_translator": false,
  "lang": "en",
  "listed_count": 1,
  "location": "",
  "name": "Michael Dropsit",
  "notifications": false,
  "profile_background_color": "C0DEED",
  "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
  "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png",
  "profile_banner_url": "https://pbs.twimg.com/profile_banners/279330988/1387392271",
  "profile_image_url": "http://pbs.twimg.com/profile_images/421677287676080128/ZObRiBf3_normal.jpeg",
  "profile_image_url_https": "https://pbs.twimg.com/profile_images/421677287676080128/ZObRiBf3_normal.jpeg",
  "profile_link_color": "0084B4",
  "profile_sidebar_border_color": "C0DEED",
  "profile_sidebar_fill_color": "DDEEF6",
  "profile_text_color": "333333",
  "profile_use_background_image": true,
  "protected": false,
  "screen_name": "MichaelDros",
  "statuses_count": 10490,
  "time_zone": "Eastern Time (US & Canada)",
  "url": null,
  "utc_offset": -18000,
  "verified": false}
}
```

Embedded Objects

- User data in the tweet is an embedded object
 - Other embedded objects, “media”, “user_mentions”, “retweeted_status” ...
- Embedded user data may be enough
- These may be stale, out of date
 - Useful, but you may not want to trust this data

Reviewing UserLookup.py code

- UserLookup.py object hierarchy



Command Line usage

- Part of the HCDE User Module

- Look in hcde/twitter

- Like the other code get a little help – run without command line parameters

```
python UserLookup.py
```

```
USAGE: python UserLookup.py -auth <appname> -user <auth_user> [-n <username>  
| -id <userid>] [-json] [-limits] [-test names|ids|both]
```

User Lookup API Specification

- Twitter Dev Site
 - <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-lookup>
- Returns full user data for up to 100 users in one request
- Two possible parameters
 - user_id
 - screen_name

Demo (setup oauth login)

```
# You should all be really used to this by now
import sys, json
from hcde.twitter.Login import Login
from hcde.twitter.UserLookup import UserLookup
from hcde.twitter.auth_settings import *
app = "HCDE530Test01"
user = "dwmcphd"
app_keys = TWITTER_APP_OAUTH_PAIR(app=app)
app_token_fname = TWITTER_APP_TOKEN_FNAME(app=app)
lg = Login(app_name=app, app_user=user, token_fname=app_token_fname)
lg.set_consumer_key(consumer_key=app_keys['consumer_key'])
lg.set_consumer_secret(consumer_secret=app_keys['consumer_secret'])
lg.login()
```

Demo (setup UserLookup)

```
# Assuming imports and lg (Login object from prior page)
ul = UserLookup()
ul.set_auth_obj(obj=lg) # all Twitter requests require auth
ul.set_user_agent(agent="safari")
ul.set_throttling(True)
# You can add up to 100 usernames or user ids
ul.add_username("ThePSF")
ul.add_username("jimmyfallon")
ul.add_username("timoreilly")
# Make the request
ul.make_request()
# like the lecture demos before, no threading, just a request
```

Demo (getting profile results)

```
# Check for messages
print ul.messages()
# Get the message from the message queue
resp = ul.get_message()
# The response is a list of dictionary items, json response was converted
print len(resp)
#
for u in resp:
    print json.dumps(u,indent=4,sort_keys=True)
```

Demo (add more users)

```
# You can add up to 100 usernames or user ids
ul.add_username("bronxzooscobra")
ul.add_username("dwmcphd")
ul.add_username("barackobama")
ul.add_username("realdonaldtrump")

# Make the next request
ul.make_request()
print ul.messages()
resp = ul.get_message()
print len(resp)
print json.dumps(resp[0],indent=4,sort_keys=True)
```

Excerpt of the profile response

■ Consider friends and followers (Feb 2018)

```
"screen_name": "ThePSF",  
"followers_count": 178750,  
"friends_count": 103,  
#  
"screen_name": "jimmyfallon",  
"followers_count": 50506331,  
"friends_count": 7801,  
#  
"screen_name": "timoreilly",  
"followers_count": 1987141,  
"friends_count": 1837,  
#  
"screen_name": "dwmcphd",  
"followers_count": 544,  
"friends_count": 177,
```

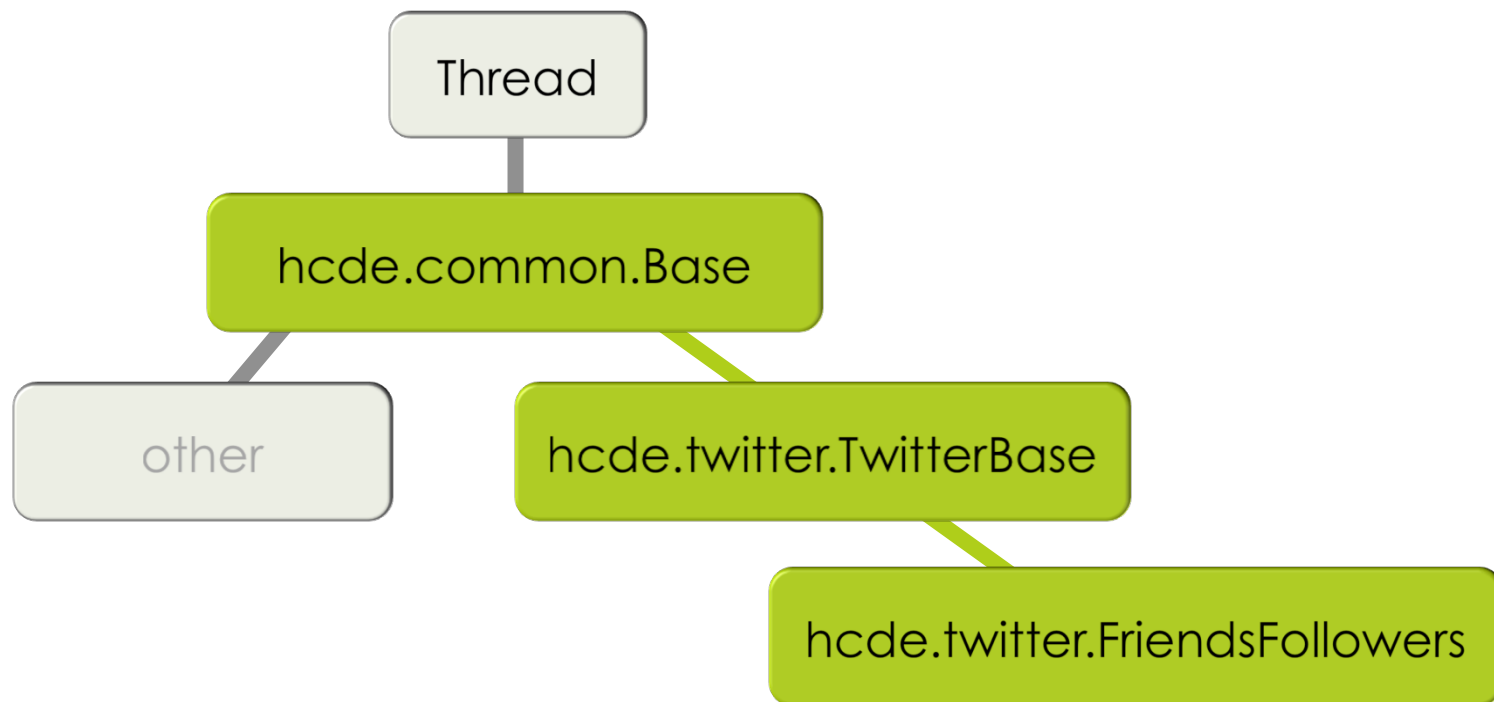
```
"screen_name": "BronxZoosCobra",  
"followers_count": 159041,  
"friends_count": 2,  
#  
"screen_name": "BarackObama",  
"followers_count": 100198777,  
"friends_count": 624597,  
#  
"screen_name": "realDonaldTrump",  
"followers_count": 48131900,  
"friends_count": 45,
```

Friends & Followers

- The user profile contains a count of friends and a count of followers
 - The counts could be useful as a simple (simplistic) measure of a users' influence
 - $\text{follower_count} / (\text{friend_count} + \text{follower_count})$
 - $\text{dwmcphd} = (544 / (177 + 544)) = 544 / 721 = 0.7545$
 - $\text{jimmyfallon} = (50506331 / 50514132) = 0.9998$
- How do you get the list of people who are friends/followers?

Reviewing FriendsFollowers.py code

- ▣ FriendsFollowers.py object hierarchy



Command Line usage

- Part of the HCDE User Module
 - Look in hcde/twitter

- Like the other code get a little help – run without command line parameters

```
python FriendsFollowers.py
```

```
USAGE: python FriendsFollowers.py -auth <appname> -user <auth_user> [-friends  
| -followers] -n <username> | -id <userid> [-count <count_per_request>] [-  
json]
```

User Lookup API Specification

- Twitter Dev Site
 - <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-ids>
 - <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-followers-ids>
 - Request by either `user_id` or `screen_name`
 - Returns up to 5000 friends/followers per response
 - Cursor to page through the friends/followers
-

Consider Rate Limits

- Our five users from before:

ThePSF "followers_count": 178750

jimmyfallon "followers_count": 50506331

timoreilly "followers_count": 1987141

dwmcphd "followers_count": 544

- Consider "ThePSF" with more than 178K followers

- $178750/5000 = 36$ requests

- 36 minutes, best case

- Assuming current Twitter limits

- 15 minute windows, 15 requests per window

Consider Rate Limits

- Consider “timoreilly” about 1.9 million followers
 - $1987141/5000 = 398$ requests
 - 398 minutes = 6.63 hours (best case)

Consider Rate Limits

- Consider “timoreilly” about 1.9 million followers
 - $1987141/5000 = 398$ requests
 - 398 minutes = 6.63 hours (best case)
- Jimmy Fallon 50.5 million followers
 - $50,506,331/5000 = 10102 = 168.35$ hours = 7.01 days

Consider Rate Limits

- Consider “timoreilly” about 1.9 million followers
 - $1987141/5000 = 398$ requests
 - 398 minutes = 6.63 hours (best case)
 - Jimmy Fallon 50.5 million followers
 - $50,506,331/5000 = 10102 = 168.35$ hours = 7.01 days
 - Barack Obama 100 million followers
 - $100,198,777/5000 = 20039.75 = 334$ hours = 13.9 days
-

Demo (setup oauth login)

```
import sys, json
from hcde.twitter.Login import Login
from hcde.twitter.FriendsFollowers import FriendsFollowers
from hcde.twitter.auth_settings import *
app = "HCDE530Test01"
user = "dwmcphd"
app_keys = TWITTER_APP_OAUTH_PAIR(app=app)
app_token_fname = TWITTER_APP_TOKEN_FNAME(app=app)
lg = Login(app_name=app, app_user=user, token_fname=app_token_fname)
lg.set_consumer_key(consumer_key=app_keys['consumer_key'])
lg.set_consumer_secret(consumer_secret=app_keys['consumer_secret'])
lg.login()
```

Demo (setup FriendsFollowers)

```
# Assuming imports and lg (Login object from prior page)
ff = FriendsFollowers()
ff.set_auth_obj(obj=lg) # all Twitter requests require auth
ff.set_user_agent(agent="ie")
# The object implements BOTH friends and follower requests
ff.set_request_type_as_friends()
#ff.set_request_type_as_followers()
ff.set_throttling(True)
ff.set_count(30)
ff.set_username("dwmcphd")
print ff.get_rate_limit()
ff.make_request()
print ff.messages()
resp = ff.get_message()
print ff.messages()
print json.dumps(resp["ids"], indent=4, sort_keys=True)
```

Caveats

- Current implementation of FriendsFollowers.py
 - Will request *all* friends/followers with the cursor
 - Friends and Followers requests have very strict rate limits
 - Max 15 requests in each 15 minute window
 - With throttling turned on (the default) code will make about 1 request per minute
 - If you know that you're requesting fewer than 75,000 then you could just go for it
 - but then you'll have to wait 15 minutes
-