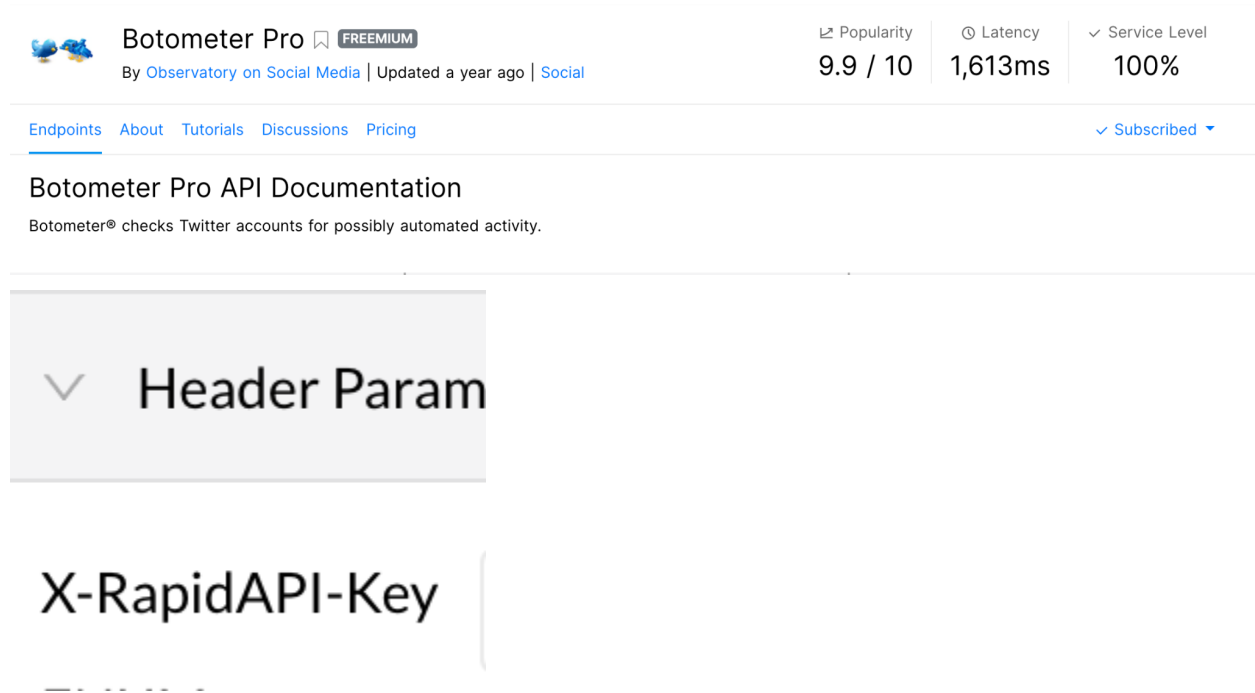


Installation

To use botometer, you need an elevated twitter api. Detailed tutorial is in README.txt in twitterapi.

Then you need a rapid key. Enter <https://rapidapi.com/OSoMe/api/botometer-pro> log in and there is rapid key



Botometer Pro FREEMIUM

By [Observatory on Social Media](#) | Updated a year ago | [Social](#)

Popularity	Latency	Service Level
9.9 / 10	1,613ms	100%

[Endpoints](#) [About](#) [Tutorials](#) [Discussions](#) [Pricing](#) [Subscribed](#)

Botometer Pro API Documentation

Botometer® checks Twitter accounts for possibly automated activity.

Header Param

X-RapidAPI-Key

Finally run

```
pip3 install botometer
```

Note if you want to use BotometerLite, you need to subscribe.

bm.py test

This code file is used to check accounts by user id or screen name.

In the file, there is a dictionary including API Key and Secret, Access Token and Secret and variable `rapidapi_key`. Change these variables to your own.

If you only check one account, put the user id or screen name into `bom.check_account`. For example,

```
result = bom.check_account('@GenshinImpact')
print(result)
```

run
python3 bm.py

Result

```
{'cap': {'english': 0.7892294694438972, 'universal': 0.7982287282125168}, 'display_scores': {'english': {'astroturf': 0.4, 'fake_follower': 0.0, 'financial': 0.0, 'other': 2.6, 'overall': 1.8, 'self_declared': 0.1, 'spammer': 0.2}, 'universal': {'astroturf': 0.4, 'fake_follower': 0.6, 'financial': 0.2, 'other': 2.9, 'overall': 1.8, 'self_declared': 0.4, 'spammer': 0.2}}, 'raw_scores': {'english': {'astroturf': 0.09, 'fake_follower': 0.18, 'financial': 0.0, 'other': 0.52, 'overall': 0.35, 'self_declared': 0.02, 'spammer': 0.03}, 'universal': {'astroturf': 0.08, 'fake_follower': 0.12, 'financial': 0.03, 'other': 0.59, 'overall': 0.35, 'self_declared': 0.07, 'spammer': 0.05}}, 'user': {'majority_lang': 'en', 'user_data': {'id_str': '1072404907230060544', 'screen_name': 'GenshinImpact'}}}
```

If you want to check more than one account, create a list of user ids and screen names and put it into bom.check_accounts_in. For example,

```
f = open("demofile2.txt", "w+")
accounts = ['@GenshinImpact', '@damien']
for result in bom.check_accounts_in(accounts):
    s = str(result) + "\n"
    f.write(s)
f.close()
```

run
python3 bm.py

Result

```
('@GenshinImpact', {'cap': {'english': 0.7892294694438972, 'universal': 0.8016}, 'display_scores': {'english': {'astroturf': 0.4, 'fake_follower': 0.0, 'financial': 0.0, 'other': 2.6, 'overall': 1.8, 'self_declared': 0.1, 'spammer': 0.2}, 'universal': {'astroturf': 0.4, 'fake_follower': 0.6, 'financial': 0.2, 'other': 2.9, 'overall': 1.8, 'self_declared': 0.4, 'spammer': 0.2}}, 'raw_scores': {'english': {'astroturf': 0.09, 'fake_follower': 0.18, 'financial': 0.0, 'other': 0.52, 'overall': 0.35, 'self_declared': 0.02, 'spammer': 0.03}, 'universal': {'astroturf': 0.08, 'fake_follower': 0.12, 'financial': 0.03, 'other': 0.59, 'overall': 0.35, 'self_declared': 0.07, 'spammer': 0.05}}, 'user': {'majority_lang': 'en', 'user_data': {'id_str': '1072404907230060544', 'screen_name': 'GenshinImpact'}}})
('@damien', {'cap': {'english': 0.7946723107742963, 'universal': 0.80534290273}, 'display_scores': {'english': {'astroturf': 0.4, 'fake_follower': 0.0, 'financial': 0.0, 'other': 2.6, 'overall': 1.8, 'self_declared': 0.1, 'spammer': 0.2}, 'universal': {'astroturf': 0.4, 'fake_follower': 0.6, 'financial': 0.2, 'other': 2.9, 'overall': 1.8, 'self_declared': 0.4, 'spammer': 0.2}}, 'raw_scores': {'english': {'astroturf': 0.09, 'fake_follower': 0.18, 'financial': 0.0, 'other': 0.52, 'overall': 0.35, 'self_declared': 0.02, 'spammer': 0.03}, 'universal': {'astroturf': 0.08, 'fake_follower': 0.12, 'financial': 0.03, 'other': 0.59, 'overall': 0.35, 'self_declared': 0.07, 'spammer': 0.05}}, 'user': {'majority_lang': 'en', 'user_data': {'id_str': '1072404907230060544', 'screen_name': 'GenshinImpact'}}})
```