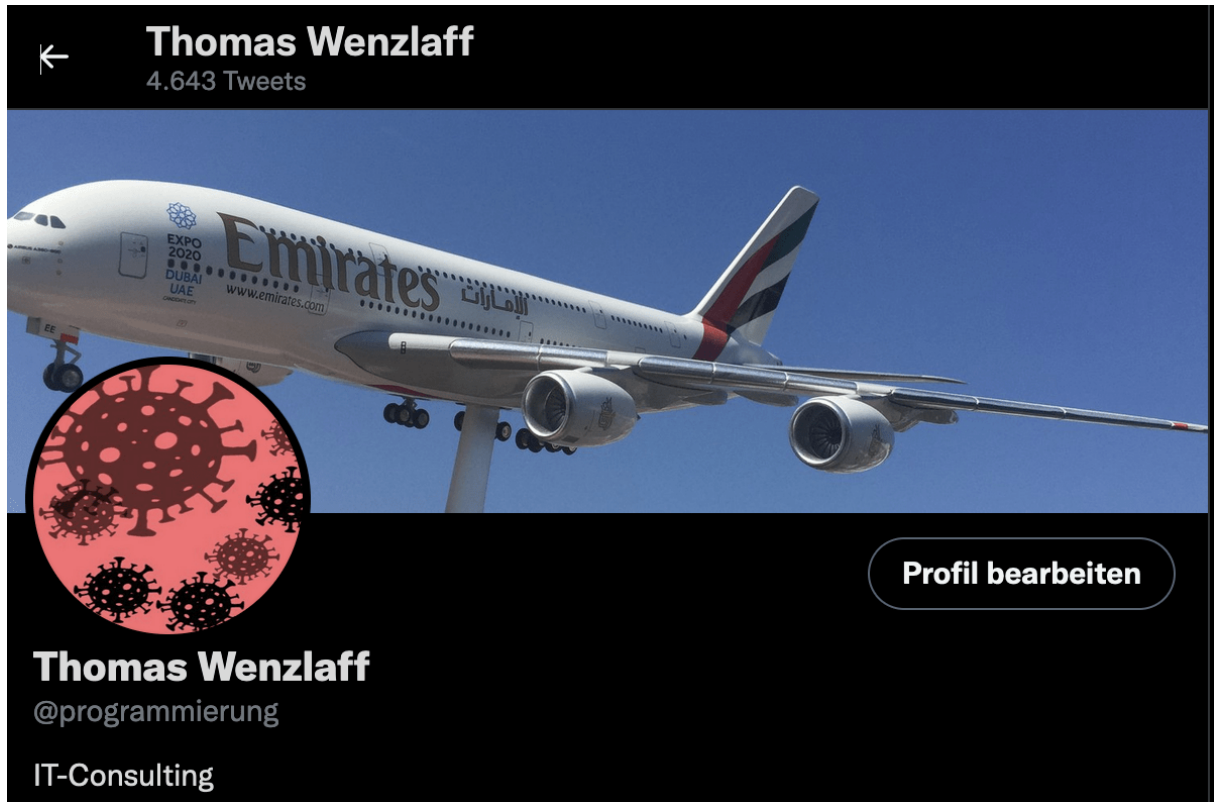


Twitter Intelligence Tool (TWINT) mit Web-Scraping-Technologie auf dem Raspberry Pi

Das Twitter Intelligence Tool (TWINT) mit Web-Scraping-Technologie kann auch auf einem Raspberry Pi installiert werden.



Mit dem Web Scraper für Twitter, mit dem man Tweets von Usern des Kurznachrichtendienstes einholen kann, ohne dabei vom Twitter API eingeschränkt zu werden läuft unter der MIT-Lizenz.

TWINT wurde in Python geschrieben, ist am April 2020 in Version 2.1.20 erschienen. Alle Funktionen gehen aber nicht mehr, weil sich die Twitter API geändert hat. Z.b. wirft ein **twint -u programmierung -following** diesen Fehler:

CRITICAL:root:twint.feed:Follow:IndexError

Aber einiges geht doch noch. Also mal installieren mit:

pip3 install twint

wenn dann nicht alles geht, hat bei mir geholfen:

- 1 sudo apt-get install libatlas-base-dev
- 2 pip3 install --upgrade -e git+https://github.com/twintproject/twint.git@origin/master#egg=twint

So ein erster Test, alle Befehle ausgeben mit `twint -h`:

```
1 TWINT - An Advanced Twitter Scraping Tool.
2
3 optional arguments:
4 -h, --help      show this help message and exit
5 -u USERNAME, --username USERNAME
6                 User's Tweets you want to scrape.
7 -s SEARCH, --search SEARCH
8                 Search for Tweets containing this word or phrase.
9 -g GEO, --geo GEO  Search for geocoded Tweets.
10 --near NEAR      Near a specified city.
11 --location       Show user's location (Experimental).
12 -l LANG, --lang LANG Search for Tweets in a specific language.
13 -o OUTPUT, --output OUTPUT
14                 Save output to a file.
15 -es ELASTICSEARCH, --elasticsearch ELASTICSEARCH
16                 Index to Elasticsearch.
17 --year YEAR      Filter Tweets before specified year.
18 --since DATE     Filter Tweets sent since date (Example: "2017-12-27
19                 20:30:15" or 2017-12-27).
20 --until DATE     Filter Tweets sent until date (Example: "2017-12-27
21                 20:30:15" or 2017-12-27).
22 --email          Filter Tweets that might have email addresses
23 --phone          Filter Tweets that might have phone numbers
24 --verified       Display Tweets only from verified users (Use with -s).
25 --csv            Write as .csv file.
26 --tabs           Separate CSV fields with tab characters, not commas.
27 --json           Write as .json file
28 --hashtags       Output hashtags in seperate column.
29 --cashtags       Output cashtags in seperate column.
30 --userid USERID  Twitter user id.
31 --limit LIMIT    Number of Tweets to pull (Increments of 20).
32 --count          Display number of Tweets scraped at the end of
33                 session.
34 --stats          Show number of replies, retweets, and likes.
35 -db DATABASE, --database DATABASE
36                 Store Tweets in a sqlite3 database.
37 --to USERNAME    Search Tweets to a user.
38 --all USERNAME   Search all Tweets associated with a user.
39 --followers      Scrape a person's followers.
40 --following      Scrape a person's follows
41 --favorites      Scrape Tweets a user has liked.
42 --proxy-type PROXY_TYPE
43                 Socks5, HTTP, etc.
44 --proxy-host PROXY_HOST
45                 Proxy hostname or IP.
46 --proxy-port PROXY_PORT
47                 The port of the proxy server.
48 --tor-control-port TOR_CONTROL_PORT
```

```
49         If proxy-host is set to tor, this is the control port
50 --tor-control-password TOR_CONTROL_PASSWORD
51         If proxy-host is set to tor, this is the password for
52         the control port
53 --essid [ESSID]   Elasticsearch Session ID, use this to differentiate
54         scraping sessions.
55 --userlist USERLIST Userlist from list or file.
56 --retweets       Include user's Retweets (Warning: limited).
57 --format FORMAT   Custom output format (See wiki for details).
58 --user-full      Collect all user information (Use with followers or
59         following only).
60 -tl, --timeline  Collects every tweet from a User's Timeline. (Tweets,
61         RTs & Replies)
62 --translate      Get tweets translated by Google Translate.
63 --translate-dest TRANSLATE_DEST
64         Translate tweet to language (ISO2).
65 --store-pandas STORE_PANDAS
66         Save Tweets in a DataFrame (Pandas) file.
67 --pandas-type [PANDAS_TYPE]
68         Specify HDF5 or Pickle (HDF5 as default)
69 -it [INDEX_TWEETS], --index-tweets [INDEX_TWEETS]
70         Custom Elasticsearch Index name for Tweets.
71 -if [INDEX_FOLLOW], --index-follow [INDEX_FOLLOW]
72         Custom Elasticsearch Index name for Follows.
73 -iu [INDEX_USERS], --index-users [INDEX_USERS]
74         Custom Elasticsearch Index name for Users.
75 --debug          Store information in debug logs
76 --resume TWEET_ID Resume from Tweet ID.
77 --videos         Display only Tweets with videos.
78 --images         Display only Tweets with images.
79 --media         Display Tweets with only images or videos.
80 --replies        Display replies to a subject.
81 -pc PANDAS_CLEAN, --pandas-clean PANDAS_CLEAN
82         Automatically clean Pandas dataframe at every scrape.
83 -cq CUSTOM_QUERY, --custom-query CUSTOM_QUERY
84         Custom search query.
85 -pt, --popular-tweets
86         Scrape popular tweets instead of recent ones.
87 -sc, --skip-certs Skip certs verification, useful for SSC.
88 -ho, --hide-output Hide output, no tweets will be displayed.
89 -nr, --native-retweets
90         Filter the results for retweets only.
91 --min-likes MIN_LIKES
92         Filter the tweets by minimum number of likes.
93 --min-retweets MIN_RETWEETS
94         Filter the tweets by minimum number of retweets.
95 --min-replies MIN_REPLIES
96         Filter the tweets by minimum number of replies.
97 --links LINKS    Include or exclude tweets containing one or more links.
98         If not specified you will get both tweets that might
```

TWINT – <http://www.wenzlaff.info>

```

99             contain links or not.
100 --source SOURCE   Filter the tweets for specific source client.
101 --members-list MEMBERS_LIST
102             Filter the tweets sent by users in a given list.
103 -fr, --filter-retweets
104             Exclude retweets from the results.
105 --backoff-exponent BACKOFF_EXPONENT
106             Specify a exponent for the polynomial backoff in case
107             of errors.
108 --min-wait-time MIN_WAIT_TIME
109             specify a minimum wait time in case of scraping limit
110             error. This value will be adjusted by twint if the
111             value provided does not satisfy the limits constraints

```

Für Analysen, natürlich nur gegen den eigenen Twitter Account absetzen. Also mal die erste Abfrage an Nutzer **programmierung** (das bin ich) mit dem Suchbegriff **mindmap**:

```
twint -u programmierung -s mindmap
```

Hier das Ergebnis:

```

1453208321893273603 2021-10-27 05:53:57 +0200 <programmierung>
#Mindmap: 6 Methods to Make Money with
Cryptocurrencies https://t.co/PXKXvki85U
1452333822066970634 2021-10-24 19:59:00 +0200 <programmierung> #crypto
#btc #bitcoin #eth #Ethereum #cryptofit #bad #token #projects #pump #dump
1 #hannover #langenhagen #mindmap #signsofbadcryptoprojects
2 from https://t.co/SKegG22Gnm https://t.co/BajmYdNCe9
3 1452121162310422532 2021-10-24 05:53:58 +0200 <programmierung>
4 #Mindmap: TOP 8 – Wie schaffe ich es, mich um mich selbst zu
5 kümmern? https://t.co/4y4LKQyurW
6 1451396403729682435 2021-10-22 05:54:02 +0200 <programmierung>
#Mindmap: 7 Anzeichen fehlender Selbstfürsorge https://t.co/gOHmsScGHp
1451018890365001729 2021-10-21 04:53:56 +0200 <programmierung>
#Mindmap: JUnit Test https://t.co/F111MkX5p0
1450973593085648898 2021-10-21 01:53:57 +0200 <programmierung>
#Mindmap: Ganache Blockchain https://t.co/nmpbqFwCmu

```

Oder mal eine Abfrage an den User **programmierung** wir suchen maximal 20 einträge mit dem Begriff **raspberry** und speichern dies in ein Json Datei:

```
twint -u programmierung -s raspberry -limit 20 -o twitter-top-20-raspberry.json -json
```

Wir bekommen dann in der json Datei diese Daten:

- 1451894525614505990 2021-10-23 14:53:24 +0200 <programmierung> ua-parser-js updates garantieren keine Sicherheit und installiert Krypto-Miner auch auf Raspberry Pi? – allow a remote attacker to obtain sensitive information or take control of the system. <https://t.co/C6uNXEiaQE>
- 1439135429303930882 2021-09-18 09:53:18 +0200 <programmierung> WordPress auf Sicherheitslücken überprüfen mit WPScan auf einem Raspberry PI unter Debian – es muss nicht immer Kali sein und “ I Still Have Faith In You“ <https://t.co/PAiz6oMPKH>
- 1421635114828713984 2021-08-01 02:53:18 +0200 <programmierung> CA-Zertifikat mit Elliptic Curve Cryptography (ECC) Key auf dem Raspberry Pi für 10 Jahre erzeugen <https://t.co/Su306g0U4r>
- 1421408785374093317 2021-07-31 11:53:57 +0200 <programmierung> Raspberry Pi Planespotter: Wieviele Flugzeuge wurden so empfangen in HAJ am Freitag? <https://t.co/C98uiZSI3k>
- 1 1421363409447178241 2021-07-31 08:53:39 +0200 <programmierung> Wie
2 können auf einem Raspberry Pi mit apt list alle möglichen Packages updates
3 angezeigt werden? <https://t.co/HoWfurI8Mj>
- 4 1419687307221819401 2021-07-26 17:53:25 +0200 <programmierung> OpenSSL
5 3.0.0 beta selbst compilieren auf einem Raspberry Pi in ca. 1-2
6 Stunden <https://t.co/sKi6SUpRf9>
- 7 1419053059162914820 2021-07-24 23:53:08 +0200 <programmierung> OpenSSL
8 nicht nur mit dem Raspberry Pi <https://t.co/tJQUjCAO1X>
- 9 1408514783645732867 2021-06-25 21:57:48 +0200 <programmierung> Wieviele
10 Flugzeuge empfängt der Raspberry Pi bei mir in EDDV (HAJ) im Durchschnitt? >
11 400 <https://t.co/dngURLaEN>
- 12 1402716556195504128 2021-06-09 21:57:42 +0200 <programmierung>
13 Auslastung des Rechners (kein Raspberry Pi) bei Brute-Force oder mit anderen
14 Worten 100% <https://t.co/A0DPN5Zu2F>
- 15 1402474902893076481 2021-06-09 05:57:28 +0200 <programmierung> SSH
16 Anmeldung an dem Raspberry Pi ohne Passwort mit Elliptische Kurve – ed25519
17 Key <https://t.co/mC4G8vxPhI>
- 18 1401170483215753216 2021-06-05 15:34:10 +0200 <programmierung> #tor
19 #darknet #raspberrypi <https://t.co/LwN3Qp3QRs> #crypto aber nicht
20 #btc #bitcoin <https://t.co/viHFabh75n>
- 21 1401161312105807877 2021-06-05 14:57:43 +0200 <programmierung> Kostenlos
eine Webseite im Darknet mit Raspberry Pi bereitstellen über Onion
Service <https://t.co/XaZkr2XeaT>
- 1399938087338319872 2021-06-02 05:57:04 +0200 <programmierung> Video:
Raspberry Pi update bzw. upgrade Prozess per BPMN
Flow <https://t.co/OCS4NwWRiW>
- 1394200412878692354 2021-05-17 09:57:36 +0200 <programmierung> Release
21.05 von rtl_433 für den Raspberry Pi veröffentlicht um Temperatursensoren und
Lufruck von Autoreifen (TPMS) uä. zu empfangen <https://t.co/uLoBEdqz7d>
- 1383600488264830977 2021-04-18 03:57:17 +0200 <programmierung> Firewall
in 10 Minuten einrichten mit UFW (Uncomplicated firewall) nicht nur auf dem
Raspberry Pi <https://t.co/zDlJrH8iSv>
- 1380233382098055173 2021-04-08 20:57:36 +0200 <programmierung> Neue
NodeRed Version 1.3.1 veröffentlicht auch für Docker auf dem Raspberry Pi (v6,
v7) <https://t.co/3H5myeWLDw>

TWINT – <http://www.wenzlaff.info>

```
1380228628248727562 2021-04-08 20:38:43 +0200 <programmierung> nice new
#nodered #docker für den #raspberrypi Version 1.3.1 https://t.co/OLomdE6DvY
1377893011565187073 2021-04-02 09:57:48 +0200 <programmierung>
Überwachung und Erkennung mit Raspberry Pi Kamera oder wie Zähle ich die
Besucher oder PKWs https://t.co/yy3WeNXD2z
1376958384151347210 2021-03-30 20:03:56 +0200 <programmierung>
Ergebnisse der Umfrage oder finde Dubletten auf dem Raspberry Pi mit
fdupes https://t.co/JtoS8nn1Sc
1373136666278379521 2021-03-20 05:57:47 +0200 <programmierung> VNC in 4
Schritte mit dem Raspberry Pi https://t.co/CU6I5kDsWs
1372910173984067585 2021-03-19 14:57:47 +0200 <programmierung> In
Pandemie wurde für den Raspberry Pi Imager eine neue Version 1.6 veröffentlicht
mit hidden Dialog https://t.co/x46bUBdUDw
```

Es geht auch eine Abfrage mit Suchbegriffen und einen Ortsbezug, z.B. Langenhagen geht:

```
twint -s bitcoin -near Langenhagen
```

Da werden dann auch alle Einträge von allen Nutzern angezeigt, und nicht nur für den eigenen Account 😊

Und auch über Python-Scripte kann man alles steuern, z.B. mein kleines `suche.py`Modul:

```
1 import twint
2
3 search = input("Eingabe des Suchbegriff:");
4 city = input("Eingabe der Stadt:")
5
6 c = twint.Config()
7 c.Search = search
8 c.Near = city
9 c.Limit = 20
10 c.Populer_tweets = True
11
12 twint.run.Search(c)
```

Es kann wie folgt gestartet werden: `python3 suche.py`

Es folgen dann diese Eingabefelder:
Eingabe des Suchbegriff:Blockchain
Eingabe der Stadt:Langenhagen

... Ausgabe des Ergebnis ... und weitere Anregungen hier.