
db2twitter Documentation

Release 0.4

Carl Chenet <chaica@ohmytux.com.org>

December 20, 2015

1	Guide	3
1.1	How to install db2twitter	3
1.2	Configure db2twitter	3
1.3	Use db2twitter	5
1.4	License	5
1.5	Authors	5
2	Indices and tables	7

db2twitter automatically extracts fields from your database, use them to feed a template of tweet and send the tweet. From 0.2 db2twitter stores already sent tweets in a sqlite3 database.

You'll find below anything you need to install, configure or run db2twitter.

1.1 How to install db2twitter

1.1.1 From PyPI

```
$ pip3 install db2twitter
```

1.1.2 From sources

- You need at least Python 3.4.
- Untar the tarball and go to the source directory with the following commands:

```
$ tar zxvf db2twitter-0.4.tar.gz  
$ cd db2twitter
```

- Next, to install db2twitter on your computer, type the following command with the root user:

```
$ python3.4 setup.py install  
$ # or  
$ python3.4 setup.py install --install-scripts=/usr/bin
```

1.2 Configure db2twitter

As a prerequisite to use db2twitter, you need a Twitter app. Log in Twitter, go to <https://apps.twitter.com>, create an app and generate the access token.

In order to configure db2twitter, you need to create a db2twitter.ini file (or any name you prefer, finishing with the extension .ini) with the following parameters:

```
[twitter]  
consumer_key=pPmJ3Bjlb2patls4r7AQW1k1l  
consumer_secret=lpj1kvnzbJxfKmeQtaQz18wm94klhaYIw5vaXq011mwId1wi1j  
access_token=1234567897-k8aN3Y5f6cfGgWhhLEuDGad1UPKHnPzkQHmP3q1  
access_token_secret=nLQwDFwZR123456789uGE6YXIukY74TmBK6JLEc123456  
tweet={} hires a {} https://www.linuxjobs.fr/jobs/{}  
hashtags=devops,linux,debian,redhat,python,java,php,mysql,postgresql  
upper_first_char=true
```

```
[database]
; use the following for PostgreSQL - you need mysql_connector_python
dbconnector=mysql+mysqlconnector
; use the following for PostgreSQL - you need psycopg2 python library
; dbconnector=postgresql+psycopg2
dbhost=localhost
database=yourdatabase
dbuser=yourdatabaseuser
dbpass=V3rYs3cr3t
databases=jobs,
jobs_rows=company_name,title,id
;jobs_sqlfilter=status=1

[sqlite]
sqlitepath=/var/lib/db2twitter/db2twitter.db

[timer]
days=mon-fri,
hours=0-11,14-17,

[circle]
last_tweets=3
each_time=2
```

For the [twitter] section:

- consumer_key: the Twitter consumer key (see your apps.twitter.com webpage)
- consumer_secret: the Twitter consumer secret key (see your apps.twitter.com webpage)
- access_token: the Twitter access token key (see your apps.twitter.com webpage)
- access_token_secret: the Twitter access token secret key (see your apps.twitter.com webpage)
- tweet: your tweet template. Should be a Python string format (see <https://docs.python.org/3/library/string.html#format-examples>)
- hashtags: a # will be added to these words in your tweets
- upper_first_char: use true if you want the first character of your tweets is upper case, false otherwise

For the [database] section:

- dbconnector: the SQLAlchemy connector to use to access your database (see examples)
- dbhost: the host where the database runs
- database: the name of the database
- dbuser: the user name to access the database
- dbpass: the password to access the database
- databases: a comma-separated list of tables to use to get data from
- jobs_rows: you should replace the jobs_rows field by a [your table]_rows field. You should have as much fields as the number of specified databases. This field contains a comma-separated name of the row to get data from
- jobs_sqlfilter: a string to pass to SQLAlchemy filter() function in order to be a new filter condition on the table you want to parse. Useful e.g if you want to ignore some rows of your table

For the [sqlite] section:

- sqlitepath: the path to the sqlite3 database

For the [timer] section:

- days: weekdays (mon for monday, thu for thursday, wed for wednesday, tue for tuesday, fri for friday, sat for saturday, sun for sunday) when db2twitter is authorized to send tweets
- hours: hours of the day (0 to 23) when db2twitter is authorized to send tweets

For the [circle] section

- last_tweets: is the number of last tweets in the sqlite database to circle around
- each_time: how many tweets to send at each execution with the circle parameter

1.3 Use db2twitter

After the configuration of db2twitter, just launch the following command:

```
$ db2twitter /path/to/db2twitter.ini
```

To use the circle mode, meaning you will alternatively tweet X tweets of your Y last tweets of your database (see the *Configure db2twitter* section), use the parameter circle:

```
$ db2twitter circle /path/to/db2twitter.ini
```

We recommend using db2twitter with cron. The following line in /etc/crontab will check for new db rows in your database every minute, build and send tweets accordingly:

```
# m h dom mon dow user  command
* * * * * db2twitter db2twitter /path/to/db2twitter.ini
0 * * * * db2twitter db2twitter circle /path/to/db2twitter.ini
```

1.4 License

This software comes under the terms of the **GPLv3+**. See the LICENSE file for the complete text of the license.

1.5 Authors

Carl Chenet <chaica@ohmytux.com>

Indices and tables

- `genindex`
- `modindex`
- `search`