feed2toot Documentation

Release 0.16

Carl Chenet <carl.chenet@ohmytux.com.org>

Dec 09, 2020

Contents

Feed2toot parses a RSS feed, extracts the last entries and sends them to Mastodon. You'll find below anything you need to install, configure or run Feed2toot.

Guide

1.1 How to install Feed2toot

1.1.1 From PyPI

$ pip3 install feed2toot

1.1.2 From sources

- You need at least Python 3.4.

- On some Linux Distribution **setuptools** package does not come with default python install, you need to install it.

- Install **PIP**:

```
$ wget https://bootstrap.pypa.io/get-pip.py -O - | sudo python3
```

- Install **setuptools** module:

```
$ wget https://bootstrap.pypa.io/ez_setup.py -O - | sudo python3
```

Alternatively, Setuptools may be installed to a user-local path:

```
$ wget https://bootstrap.pypa.io/ez_setup.py -O - | python3 - --user
```

- Untar the tarball and go to the source directory with the following commands:

```
$ tar zxvf feed2toot-0.16.tar.gz
$ cd feed2toot
```

- Next, to install Feed2toot on your computer, type the following command with the root user:

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

1.2 Configure Feed2toot

1.2.1 Create credentials for Mastodon

As a prerequisite to use Feed2toot, you need to authorize a Mastodon app for your account.

Just use the script register_feed2toot_app to register the feed2toot app for your account.:

```
$ ./register_feed2toot_app

This script generates the Mastodon application credentials for Feed2toot.
feed2toot_clientcred.txt and feed2toot_usercred.txt will be written
in the current directory: /home/me/feed2toot.
WARNING: previous files with the same names will be overwritten.

A connection is also initiated to create the application.
Your password is *not* stored.

Mastodon instance URL (defaults to https://mastodon.social): https://framapiaf.org
Mastodon login: toto@titi.com
Mastodon password:

The app feed2toot was added to your Preferences=>Accounts=>Authorized apps page.
The file feed2toot_clientcred.txt and feed2toot_usercred.txt were created in the
→current directory.
```

As described above, two files were created. See the *Use register_feed2toot_app* section for more options for register_feed2toot_app.

1.2.2 Create Feed2toot configuration

After using register_feed2toot_app, you'll need the credentials in the feed2toot configuration.

In order to configure Feed2toot, you need to create a feed2toot.ini file (or any name you prefer, finishing with the extension .ini) with the following parameters:

```
[mastodon]
instance_url=https://mastodon.social
; Here you need the two files created by register_feed2toot_app
user_credentials=/etc/feed2toot/credentials/feed2toot_usercred.txt
client_credentials=/etc/feed2toot/credentials/feed2toot_clientcred.txt
; Default visibility is public, but you can override it:
; toot_visibility=unlisted

[cache]
cachefile=/var/lib/feed2toot/feed2toot.db
cache_limit=10000

[lock]
```

```
lock_file=/var/lock/feed2toot.lock
lock_timeout=3600

[rss]
uri=https://www.journalduhacker.net/rss
uri_list=/etc/feed2toot//rsslist.txt
toot={title} {link}
; toot_max_len=125
title_pattern=Open Source
title_pattern_case_sensitive=true
no_uri_pattern_no_global_pattern=true
; ignore_ssl=false

[hashtaglist]
several_words_hashtags_list=/etc/feed2toot/hashtags.txt
; no_tags_in_toot=false

[feedparser]
accept_bozo_exceptions=true

[media]
custom=/var/lib/feed2toot/media/logo.png
```

For the [mastodon] section:

- instance_url: the url of your Mastodon instance

- user_credentials: a file with the user credentials, generated by the command register_feed2toot_app

- client_credentials: a file with the client credentials, generated by the command register_feed2toot_app

- toot_visibility: any of the valid options for the *visibility* field here. Default is *public*, but *unlisted* prevents flooding the instance's public timeline (which is more polite).

For the [cache] section:

- cachefile: the path to the cache file storing ids of already tooted links. Absolute path is mandatory. This file should always use the .db extension.

- cache_limit: length of the cache queue. defaults to 100.

For the [lock] section (starting from version 0.11):

- lock_file: lock to stop any other feed2toot instance to run at the same time. Default is ~/.config/feed2toot.lock

- lock_timeout: automatically remove the lock if the datetime in the lock file is greater than n seconds. Default is 3600 seconds.

For the [rss] section:

- uri: the url of the rss feed to parse

- uri_list: a path to a file with several adresses of rss feeds, one by line. Absolute path is mandatory.

- toot: format of the toot you want to post. It should use existing entries of the RSS fields like {title} or {link}. Launch it with this field empty to display all available entries. If you want to shorten the size of a field, you can use the syntax {summary:.100} to cut the field "summary" of the rss feed after the first 100 characters (starting from version 0.10). To add new lines you can use n (starting from version 0.14)

- toot_max_len: the max length of a toot can be defined here. If the toot size is longer, the toot is truncated and "..." added at the end. Defaults is 500 characters.

- {one field of the rss feed}_pattern: takes a string representing a pattern to match for a specified field of each rss entry of the rss feed, like title_pattern or summary_pattern.

- {one field of the rss feed}_pattern_case_sensitive: either the pattern matching for the specified field should be case sensitive or not. Default to true if not specified.

- no_uri_pattern_no_global_pattern: don't apply global pattern (see above) when no pattern-by-uri is defined in the uri_list. Allows to get all entries of a rss in the uri_list because no pattern is defined so we match them all. Defaults to false, meaning the global patterns will be tried on every rss in the uri_list NOT HAVING specific patterns and so ONLY entries from the specific uri in the uri_list matching the global patterns will be considered.

- addtags: add the tags from the rss feed at the end of the toot. Defaults to true.

- ignore_ssl: when the uri or uri_list contains an https url with an invalid certificate (e.g an expired one), feed2toot will be unable to get rss content. This option allows to bypass the ssl security to catch the rss content. Defaults to false.

For the [hashtaglist] section:

- several_words_hashtags_list: a path to the file containing hashtags in two or more words. Absolute path is mandatory. By default Feed2toot adds a # before every words of a hashtag. See documentation below for an example of this file.

- no_tags_in_toot: stop hash tags to be added at the toot. Defaults to false.

for the [feedparser] section:

- accept_bozo_exceptions: If set to true, feed2toot will accept malformed feeds, which are rejected by default.

For the [media] section:

- custom: the path to a media (should be supported by Mastodon) to be posted with every Mastodon post.

1.3 Example of the list of hash tags

The list of hash tags is a simple text file with one hash tag composed by several words on a single line:

```
free software community
open-source
```

Instead of having #free #software #community or #open-source in the resulting toot, you will have #freesoftwarecommunity and #opensource. You only have to identify the hash tags you frequently use in your RSS feeds and put them in this file to have well formatted hash tags in your toots.

1.4 List of rss feeds

1.4.1 Simple list of rss feeds

With the parameter **uri_list**, you can define a list of uri to use. Feed2toot is able to match specific patterns for each of the rss feeds from this list. Consider the following rss section of the configuration file:

```
[rss]
uri_list=/home/john/feed2toot/rsslist.txt
toot={title} {link}
```

Now let's have a look at the =/home/john/feed2toot/rsslist.txt file:

```
https://www.journalduhacker.net/rss
https://carlchenet.com/feed
```

Each line of this file is a url to a rss feed. Pretty simple.

1.4.2 Display the name of the feed in the toots

If you want to display the name of the feed in the resulting toot, you can do so by giving it a name with the following syntax:

```
Le journal du hacker <https://www.journalduhacker.net/rss/>
```

Then in the *toot* configuration, you can use the *{feedname}* syntax, which will be replaced by the actual name of the feed.

1.4.3 Match specific patterns of rss feeds in the uri_list files

You can use specific pattern matching for uri in the uri_list file to filter some of the rss entries of a rss feed. Lets modify the previous file:

```
https://www.journalduhacker.net/rss|title|hacker,psql
https://carlchenet.com/feed|title|gitlab
```

Each line of this file starts with an uri, followed by a pipe (|), followed by the name of the available section to parse (see below), again followed by a pipe (|), followed by patterns, each pattern being separated from the other one by a semi-colon (,).

In the example file above wee get every rss entries from the feed available at https://www.journalduhacker.net/rss where a substring in the title section of this entry matches either "hacker" or "psql". Specific patterns are not case sensitive. For the second line, we match every rss entries from the feed available at https://carlchenet.com/feed where a substring in the title section of this entry matches "gitlab".

1.4.4 Consider every entries of a rss feed from a uri in the uri_list file

It is possible to get all entries from a rss feed available in the uri_list file. You need an option to deactivate the global pattern matching for uri in the uri_list NOT having specific patterns:

```
[rss]
...
no_uri_pattern_no_global_pattern=true
```

In you rsslist.txt, just don't give anything else than the needed feed url to get all the entries:

```
https://www.journalduhacker.net/rss|title|hacker,psql
https://carlchenet.com/feed|title|gitlab
https://blog.linuxjobs.fr/feed.php?rss
```

The last line of the file above only has the url of a rss feed. All entries from this feed will be tooted.

1.5 Use Feed2toot

After the configuration of Feed2toot, just launch the following command:

```
$ feed2toot -c /path/to/feed2toot.ini
```

1.5.1 Run Feed2toot on a regular basis

Feed2toot should be launched on a regular basis in order to efficiently send your new RSS entries to Mastodon. It is quite easy to achieve by adding a line to your user crontab, as described below:

```
@hourly feed2toot -c /path/to/feed2toot.ini
```

will execute feed2toot every hour. Or without the syntactic sugar in the global crontab file /etc/crontab:

```
0 * * * * johndoe feed2toot -c /path/to/feed2toot.ini
```

1.5.2 Test option

In order to know what's going to be sent to Mastodon without actually doing it, use the **–dry-run** option:

```
$ feed2toot --dry-run -c /path/to/feed2toot.ini
```

1.5.3 Debug option

In order to increase the verbosity of what's Feed2toot is doing, use the **–debug** option followed by the level of verbosity see [the the available different levels](https://docs.python.org/3/library/logging.html):

```
$ feed2toot --debug -c /path/to/feed2toot.ini
```

1.5.4 Populate the cache file without posting toots

Starting from 0.8, Feed2toot offers the **–populate-cache** command line option to populate the cache file without posting to Mastodon:

```
$ feed2toot --populate-cache -c feed2toot.ini
populating RSS entry https://www.journalduhacker.net/s/65krkk
populating RSS entry https://www.journalduhacker.net/s/co2es0
populating RSS entry https://www.journalduhacker.net/s/la2ihl
populating RSS entry https://www.journalduhacker.net/s/stfwtx
populating RSS entry https://www.journalduhacker.net/s/qq1wte
populating RSS entry https://www.journalduhacker.net/s/y8mzrp
populating RSS entry https://www.journalduhacker.net/s/ozjqv0
populating RSS entry https://www.journalduhacker.net/s/6ev8jz
populating RSS entry https://www.journalduhacker.net/s/gezvnv
populating RSS entry https://www.journalduhacker.net/s/lqswmz
```

1.5.5 How to display available sections of the rss feed

Starting from 0.8, Feed2toot offers the **–rss-sections** command line option to display the available section of the rss feed and exits:

```
$ feed2toot --rss-sections -c feed2toot.ini
The following sections are available in this RSS feed: ['title', 'comments', 'authors
→', 'link', 'author', 'summary', 'links', 'tags', id', 'author_detail', 'published'].
```

1.5.6 Using syslog

Feed2toot is able to send its log to syslog. You can use it with the following command:

```
$ feed2toot --syslog=WARN -c /path/to/feed2toot.ini
```

1.5.7 Limit number of rss entries published at each execution

If you want to limit the number of rss entries published at each execution, you can use the –limit CLI option.

> $ feed2toot –limit 5 -c /path/to/feed2toot.ini

The number of posts to Mastodon will be at 5 posts top with this CLI option.

1.6 Use register_feed2toot_app

You need a Mastodon app associated to a user on the Mastodon instance. The script register_feed2toot_app will create an app for Feed2toot and upload it on the specified Mastodon instance.

Primary usage

```
$ register_feed2toot_app
```

Possible CLI options:

- use the **–client-credentials-file** option to change the filename in which the client credentials are stored (defaults to feed2toot_clientcred.txt)

- use the **–user-credentials-file** option to change the filename in which the user credentials are stored (defaults to feed2toot_usercred.txt)

- use the **–name** to change the Mastodon app name (defaults to feed2toot)

Example with full options and full output:

```
$ ./register_feed2toot_app --user-credentials-file f2tusercreds.txt --client-
→credentials-file f2tclientcreds.txt --name f2t

This script generates the Mastodon application credentials for Feed2toot.
f2tclientcreds.txt and f2tusercreds.txt will be written
in the current directory: /home/me/feed2toot/scripts.
WARNING: previous files with the same names will be overwritten.

A connection is also initiated to create the application.
Your password is *not* stored.

Mastodon instance URL (defaults to https://mastodon.social): https://framapiaf.org
Mastodon login: toto@titi.com
Mastodon password:
```

```
The app f2t was added to your preferences=>authorized apps page.
The file f2tclientcreds.txt and f2tusercreds.txt were created in the current␣
→directory.
```

1.7 Plugins

Feed2toot supports plugins. Plugins offer optional features, not supported by default. Optional means you need a dedicated configuration and sometimes a dedicated external dependencies. What you need for each module is specified below.

1.7.1 InfluxDB

The InfluxDB plugin allows to store already published toots in a InfluxDB database.

Install the InfluxDB plugin

To install Feed2toot with the InfluxDB plugin, execute the following command.

From scratch:

```
# pip3 install feed2toot[influxdb]
```

Upgrading from a previous version, execute the followin command:

```
# pip3 install feed2toot[influxdb] --upgrade
```

Configuration

Below is the block of configuration to add in your feed2toot.ini:

```
[influxdb]
;host=127.0.0.1
;port=8086
user=influxuser
pass=V3ryS3cr3t
database=influxdb
measurement=toots
```

- host: the host where the influxdb instance is. Defaults to 127.0.0.1

- port: the port where the influxdb instance is listening to. Defaults to 8086

- user: the user authorized to connect to the database. Mandatory (no default)

- pass: the password needed to connect to the database. Mandatory (no default)

- database: the name of the influxdb database to connect to. Mandatory (no default)

- measurement: the measurement to store the value into. Mandatory (no default)

1.8 License

This software comes under the terms of the **GPLv3+**. It was previously under the **MIT** license. See the LICENSE file for the complete history of the license and the full text of the past and current licenses.

1.9 Authors

Carl Chenet <carl.chenet@ohmytux.com>

CHAPTER 2

Indices and tables

- genindex
- modindex
- search