
OneBot Documentation

Release 0.1.0

Thom Wiggers

Aug 23, 2018

Contents

1	OneBot	3
1.1	Features	3
2	Installation	5
3	Usage	7
3.1	Configuration	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	1.3.2 (2018-08-23)	15
6.2	1.3.1 (2018-08-23)	15
6.3	1.3.0 (2018-08-13)	15
6.4	1.2.1 (2018-03-05)	15
6.5	1.2.0 (2017-10-31)	15
6.6	1.1.0 (2017-04-01)	16
6.7	1.0.0 (2016-07-17)	16
6.8	0.1.0 (2015-??)	16
6.9	0.0.0 (2014-05-21)	16
7	Indices and tables	17

Contents:

 OneBot is an ircbot based on irc3

- Free software: BSD license
- Documentation: <http://onebot.rtfid.org>.

1.1 Features

- Database plugin
- Last.FM plugin
- Trakt.TV plugin
- Advanced user system
- Advanced ACL
- URL information
- Utility commands

These features are available through plugins.

CHAPTER 2

Installation

To run OneBot you need at least Python 3.4.

At the command line:

```
$ pip install onebot
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv onebot  
$ pip install onebot
```

You should now have the `onebot` command available.

Use the `onebot` command to run stuff.

3.1 Configuration

You can use this sample configuration file:

```
[bot]
nick=OneBotTest
username=TwoBot
host=my.irc.network

port=6697
ssl=true
# uncomment this if you don't want to check the certificate
# ssl_verify = CERT_NONE

includes=
  irc3.plugins.uptime
  onebot.plugins.acl
  onebot.plugins.lastfm
  onebot.plugins.trakt
  onebot.plugins.botui
  onebot.plugins.whatcd
  irc3.plugins.autojoins
  onebot.plugins.urlinfo
# add your plugins here

autojoins=
  ${hash}channel

# see http://irc3.readthedocs.io/en/latest/plugins/storage.html#module-irc3.plugins.
↪ storage
```

(continues on next page)

(continued from previous page)

```
#storage = unix:///var/run/redis/redis.sock?db=10
storage = shelve:///tmp/test.shelf

# Plugin settings are noted as follows
[irc3.plugins.command]
cmd = .
guard=onebot.plugins.acl.user_based_policy

[onebot.plugins.users]
identify_by = mask

[onebot.plugins.acl]
# Pre-seed acl
superadmin=me@my.awesome.host

[onebot.plugins.lastfm]
api_key = myapikey
api_secret = myapikey

[onebot.plugins.trakt]
client_id = myclientid
```

Most options should mostly speak for themselves. Note that the plugin settings are (obviously) plugin-dependant. You should consult the manual for each plugin to figure out what you need to set.

This project is based on *irc3*. *irc3* plugins are compatible with OneBot. Some OneBot plugins make use of *irc3* modules. See the module `onebot.plugin`. You should consult the [irc3 documentation](#) for information about *irc3* plugins.

`onebot .run` (*argv=None*)

Run OneBot from a config file

Usage: `onebot [options] <config>...`

Options:

<code>--logdir DIRECTORY</code>	Log directory to use instead of stderr
<code>--logdate</code>	Show datetimes in console output
<code>-r, --raw</code>	Show raw IRC log on the console
<code>-v, --verbose</code>	Increase verbosity
<code>-d, --debug</code>	Add debug commands/utils

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/thomwiggers/onebot/issues>.

If you are reporting a bug running an instance, please include:

- Your operating system name and version.
- Python version and versions of dependencies (*pip freeze* output)
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

If you are reporting a bug using an IRC plugin or command, include the output.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

OneBot could always use more documentation, whether as part of the official OneBot docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/thomwiggers/onebot/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *onebot* for local development.

1. Fork the *onebot* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/onebot.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv onebot
$ cd onebot/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 onebot tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in `README.rst`.
3. The pull request should work for Python 3.4 and 3.5. Check https://travis-ci.org/thomwiggers/onebot/pull_requests and make sure that the tests pass for all supported Python versions.
4. Don't forget to add yourself to `CONTRIBUTORS.rst`!

5.1 Development Lead

- Thom Wiggers <thom@thomwiggers.nl>

5.2 Contributors

- Jim Rollenhagen <jim@jimrollenhagen.com>
 - Unit test refactor (PR #1)
- Joost Rijneveld <joost@joostrijneveld.nl>
 - PSA plugin (PR #32)
- Mathew ‘Mattbox’
 - Musicbrainz tags, PSA tests, plugin reload (PR #36)
- Gerdriaan Mulder ‘@mrngm’
 - Change urlinfo character limit to 320, helps twitter urls (PR #38)

6.1 1.3.2 (2018-08-23)

- Actually release the below fix

6.2 1.3.1 (2018-08-23)

- Add *Compatible: curl* to *User-Agent* for *urlinfo.py* (contributed by [@mrngm PR #39](#))

6.3 1.3.0 (2018-08-13)

- Support Python 3.7 (Requires *irc3* ≥ 1.1)

6.4 1.2.1 (2018-03-05)

- Change *urlinfo* character limit to 320, helps twitter urls (contributed by [@mrngm PR #38](#))

6.5 1.2.0 (2017-10-31)

- Truncate too long `<title>` tags in *urlinfo* output.
- Drop Python 3.3, 3.4 support

6.6 1.1.0 (2017-04-01)

- PSA Plugin (contributed by @joostrijneveld: [PR #32](#))
- Various contributions by @Mattbox: [PR #36](#) * Use musicbrainz tags instead of last.fm * Tests for PSA plugin * reload command
- Remove `compare` command as it's dead on Last.fm
- Remove `mbid` lookups on Last.fm as they're broken

6.7 1.0.0 (2016-07-17)

I get around to finally posting this shit.

6.8 0.1.0 (2015-??)

First production usage

6.9 0.0.0 (2014-05-21)

Start of development

CHAPTER 7

Indices and tables

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