
Modem Scraper Documentation

Release 0.2.0

Michael Johnson

Jul 02, 2017

Contents

1	Modem Scraper	3
1.1	Features	3
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.2.0 (2017-06-30)	15
6.2	0.1.0 (2017-06-29)	15
7	Indices and tables	17

Contents:

CHAPTER 1

Modem Scraper

Modem scraper retrieves important information from Surfboard-like modem status pages. I developed this so that I could write a Zabbix agent to monitor the status of my Charter SB6141 modem.

So far, it has been tested on the SB6141 running Charter Communication's firmware. If you would like to contribute your modem's status pages for addition, please open an issue.

- Free software: GNU General Public License v3
- Documentation: <https://modemscraper.readthedocs.io>.

1.1 Features

Modem scraper can fetch information from a supported modem's status page, including:

- operational status
- channel statistics, both upstream and downstream
- logs

2.1 Stable release

To install Modem Scraper, run this command in your terminal:

```
$ pip install modemscraper
```

This is the preferred method to install Modem Scraper, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for Modem Scraper can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/nightexcessive/modemscraper
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/nightexcessive/modemscraper/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

To use Modem Scraper in a project:

```
import modemscraper
```


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/nightexcessive/modemscraper/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

Modem Scraper could always use more documentation, whether as part of the official Modem Scraper 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/nightexcessive/modemscraper/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 *modemscraper* for local development.

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

```
$ git clone git@github.com:your_name_here/modemscraper.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 modemscraper
$ cd modemscraper/
$ 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 *pylint* and the tests, including testing other Python versions with *tox*:

```
$ pylint modemscraper tests
$ python setup.py test or py.test
$ tox
```

To get *pylint* 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/nightexcessive/modemscraper/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_modemscraper
```


5.1 Development Lead

- Michael Johnson <michael@johnson.computer>

5.2 Contributors

None yet. Why not be the first?

6.1 0.2.0 (2017-06-30)

- Added parser module supported by Pint
- Added CI support + coverage information with Coveralls

6.2 0.1.0 (2017-06-29)

- First release on PyPI.

CHAPTER 7

Indices and tables

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