Python simple arp table reader Documentation

Release 0.0.1

David Francos

Nov 17, 2017

Contents

Contents:

CHAPTER 1

Python simple arp table reader

Python simple arp table reader

- Free software: BSD license
- Documentation: https://python_arptable.readthedocs.org.

1.1 Features

- Export ARP Table on linux as a dictionary.

1.2 Usage

arp_table exports two main things:

- ARPTABLE constant, as it was when the package was imported
- get_arp_table() function, returning latest arp table.

```
>>> from arp_table import ARPTABLE
>>> print(ARPTABLE)

[{'Device': 'eth0',
  'Flags': '0x2',
  'HW address': '00:12:79:d2:b9:67',
  'HW type': '0x1',
  'IP address': '10.3.0.1',
  'Mask': '*'}]

>>> from arp_table import get_arp_table
>>> print(get_arp_table())

[{'Device': 'eth0',
```

```
  'Flags': '0x2',
  'HW address': '00:12:79:d2:b9:67',
  'HW type': '0x1',
  'IP address': '10.3.0.1',
  'Mask': '*'}]
```

1.3 Disclaimer

This is a small, simple yet quite useful library. I've seen a lot of posts of people struggling with arp-table reading in python.

With this library, you can just

```
pip install arp_table
```

and enjoy!

CHAPTER 2

Installation

At the command line:

```
$ easy_install python_arptable
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv python_arptable
$ pip install python_arptable
```

CHAPTER 3

Usage

To use Python simple arp table reader in a project:

```
import python_arptable
```

CHAPTER 4

Contributing

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/XayOn/python_arptable/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" 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

Python simple arp table reader could always use more documentation, whether as part of the official Python simple arp table reader 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/XayOn/python_arptable/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 *python_arptable* for local development.

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

```
$ git clone git@github.com:your_name_here/python_arptable.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 python_arptable
$ cd python_arptable/
$ 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 python_arptable 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 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/XayOn/python_arptable/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_python_arptable
```

CHAPTER 5

Credits

5.1 Development Lead

- David Francos <me@davidfrancos.net>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

CHAPTER 7

0.0.1 (2015-06-14)

- First release on PyPI.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search