
Pytar

Release 0.1.0.dev2

Douglas Miranda

Sep 27, 2017

Contents

1	Basics	1
1.1	(Experimental) Pytar	1
1.2	Installation	2
1.3	Usage	2
2	Project Info	3
2.1	Contributing	3
2.2	Credits	5
2.3	History	5
3	Indices and tables	7

NOTE: If you like this idea, you really should try the `unp`.

(Experimental) Pytar

Cause I'll never remember a valid tar command. Boom! I don't know if its a good or bad idea, but I'm doing anyway.

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

Features

Well, for now:

- Extracting tar files without having to remember those commands.

```
# pytar $ACTION $TARGET
$ pytar extract my-tar-file.tar
```

Install

- Using PIP.

```
$ pip install pytar --pre
```

See `-pre` flag on PIP documentation.

Installation

At the command line:

```
$ easy_install pytar
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv pytar
$ pip install pytar
```

But, for now, we are on pre-alpha version, so:

```
$ pip install pytar --pre
```

See [-pre flag](#) on PIP documentation.

Usage

To use Pytar in a project:

```
from pytar.pytar import pytar_extract
pytar_extract('my-tar-file.tar')
```

To use as a console app:

```
# pytar $ACTION $TARGET
$ pytar extract my-tar-file.tar
```

Another examples

Extracting to a different directory:

```
# pytar $ACTION $TARGET $EXTRACT_PATH
$ pytar extract my-tar-file.tar ../extract/path/
```

(The contents from *my-tar-file.tar* will be extracted to *../extract/path/* directory.)

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:

Types of Contributions

Report Bugs

Report bugs at <https://github.com/douglasmiranda/pytar/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.

Fix Bugs

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

Implement Features

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

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/douglasmiranda/pytar/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 :)

Get Started!

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

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

```
$ git clone git@github.com:your_name_here/pytar.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 pytar
$ cd pytar/
$ 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 pytar 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.

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, and 3.3, and for PyPy. Check https://travis-ci.org/douglasmiranda/pytar/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

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

Credits

Development Lead

- Douglas Miranda <douglasmirandasilva@gmail.com>

Contributors

None yet. Why not be the first?

History

0.1.0.dev2 (2013-10-11)

- Add “extract path” (issue #2)

0.1.0.dev1 (2013-10-03)

- Basic unicode issues fixed.
- Docs Info.

0.1.0.dev0 (2013-10-02)

- First release on PyPI.

CHAPTER 3

Indices and tables

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