pyrobase Documentation

Release 0.6.1

pyroscope

Feb 23, 2020

Contents

pyrobase assembles general Python helper functions and classes that can be applied to any project. That includes some additional tasks for the Paver build tool, an improved xmlrpc2scgi module, unit test helpers, and generic base modules for various domains.

Project Links

GitHub https://github.com/pyroscope/pyrobase#readme

PyPI http://pypi.python.org/pypi/pyrobase/

API docs https://pyrobase.readthedocs.io/en/latest/api.html

OpenHub https://www.openhub.net/p/pyrobase

To get in contact and share your experiences with other users of PyroScope, join the pyroscope-users mailing list or the inofficial `##rtorrent` channel on `irc.freenode.net`.

CHAPTER 1

Contents of This Manual

1.1 API Documentation

This is the full `pyrobase` API documentation, generated from source.

1.1.1 Packages & Modules

pyrobase package

Subpackages

pyrobase.io package

Submodules

pyrobase.io.http module

pyrobase.io.xmlrpc2scgi module

Module contents

pyrobase.paver package

Submodules

pyrobase.paver.documentation module

1.2 Contributing Guidelines

See contribution-guide.org for the basics on contributing to an open source project.

1.2.1 Reporting an Issue, or Requesting a Feature

Any defects and feature requests are managed using GitHub's *issue tracker*. If you never opened an issue on GitHub before, consult the Mastering Issues guide.

Before creating a bug report, please read *contribution-guide.org*'s Submitting Bugs.

1.2.2 Creating a Work Directory

First, check out the source:

```
mkdir -p ~/src
git clone https://github.com/pyroscope/pyrobase.git ~/src/pyrobase
cd $_
```

You are strongly encouraged to build within a virtualenv, call the provided script `bootstrap.sh` to create one in your working directory:

```
PYTHON=python3 ./bootstrap.sh
. .env
```

1.2.3 Common Development Tasks

Here are some common project tasks:

```
pytest          # Run unit tests
inv docs -o     # Build documentation and show in browser
inv lint        # Check code quality
inv cov         # Run unit tests & show coverage report
tox             # Run unit tests in various test environments (multiple Python␣
↪versions)
```

1.2.4 Performing a Release

1. Check for and fix `pylint` violations:

   ```
   paver lint -m
   ```

2. Verify `debian/changelog` for completeness and the correct version, and bump the release date:

   ```
   dch -r
   ```

3. Check Travis CI status at https://travis-ci.org/pyroscope/pyrobase

4. Remove 'dev' version tagging from `setup.cfg`, and perform a release check:

   ```
   sed -i -re 's/^(tag_[a-z ]+=)/##\1/' setup.cfg
   paver release
   ```

5. Commit and tag the release:

   ```
   git status  # check all is committed
   tag="v$(dpkg-parsechangelog | grep '^Version:' | awk '{print $2}')"
   git tag -a "$tag" -m "Release $tag"
   ```

6. Build the final release and upload it to PyPI:

```
paver dist_clean sdist bdist_wheel
twine upload dist/*.{zip,whl}
```

1.3 Indices & Tables

- genindex
- modindex
- search