
proctor

unknown

Feb 14, 2022

TABLE OF CONTENTS

1	proctor	3
1.1	proctor package	3
2	Changelog for proctor	5
2.1	Unreleased	5
2.2	0.2.2 - 2022-02-14	5
2.3	0.2.1 - 2022-02-08	5
2.4	0.2.0 - 2021-12-23	5
2.5	[0.1.0](https://github.com/python-boltions/proctor/releases/tag/0.1.0)	6
3	Contributing	7
3.1	How to submit feedback?	7
3.2	Developer's Guide	7
3.3	New Releases	8
4	proctor	9
4.1	Installation	9
4.2	Useful Links	9
5	Indices and Tables	11
	Python Module Index	13
	Index	15

process doctor (PROCTOR)... Utilities related to process / thread management.

1.1 proctor package

proctor package

process doctor (PROCTOR)... Utilities related to process / thread management.

```
class Process(popen, *, timeout=15)
```

Bases: object

A wrapper around a subprocess.Popen(...) object.

Examples

```
>>> from subprocess import PIPE, Popen
```

```
>>> echo_factory = lambda x: Popen(["echo", x], stdout=PIPE)
```

```
>>> echo_popen = echo_factory("foo")
>>> echo_proc = Process(echo_popen)
>>> echo_proc.out
'foo'
```

```
>>> echo_popen = echo_factory("bar")
>>> out, _err = Process(echo_popen)
>>> out
'bar'
```

Parameters

- **popen** (*Popen*) –
- **timeout** (*float* | *None*) –

```
to_error(*, up=0)
```

Converts a Process object into an Err(...) object..

Parameters **up** (*int*) –

Return type Err[*Process*, *ErisError*]

command_exists(*cmd*)

Returns True iff the shell command *cmd* exists.

Parameters *cmd* (str) –

Return type bool

safe_popen(*cmd_parts*, *, *up=0*, *timeout=15*, ***kwargs*)

Wrapper for subprocess.Popen(...).

Return type Result[*Process*, ErisError]

Returns

Ok(*Process*) if the command is successful. OR

Err(ErisError) otherwise.

Parameters

- **cmd_parts** (*Iterable[str]*) –
- **up** (*int*) –
- **timeout** (*float | None*) –
- **kwargs** (*Any*) –

unsafe_popen(*cmd_parts*, *, *timeout=15*, ***kwargs*)

Wrapper for subprocess.Popen(...)

You can use `unsafe_popen()` instead of `safe_popen()` when you don't care whether or not the command succeeds.

Return type *Process*

Returns A *Process*(...) object.

Parameters

- **cmd_parts** (*Iterable[str]*) –
- **timeout** (*float | None*) –
- **kwargs** (*Any*) –

CHANGELOG FOR PROCTOR

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#), and this project adheres to [Semantic Versioning](#).

2.1 Unreleased

No notable changes have been made.

2.2 0.2.2 - 2022-02-14

2.2.1 Fixed

- Fix log message crash.

2.3 0.2.1 - 2022-02-08

2.3.1 Fixed

- Fix type of 'timeout' kwarg.

2.4 0.2.0 - 2021-12-23

2.4.1 Changed

- *BREAKING CHANGE*: Upgrade eris dependency to version 0.2.0.

2.5 [0.1.0](<https://github.com/python-bolttons/proctor/releases/tag/0.1.0>)

2.5.1 Miscellaneous

- Port proctor library from (original) `bugyi-lib` library.
- First Release.

CONTRIBUTING

3.1 How to submit feedback?

The best way to submit feedback is to [file an issue](#).

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.

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 :).

3.2 Developer's Guide

3.2.1 Basic Usage

Before making a PR please run the following

- Optional one time setup: run `make use-docker` if you need to build/test this with docker
- `make lint` to check for any format or convention issues
- `make test` to run all tests

3.2.2 How do I ...

3.2.3 Badges

Every badge shown below corresponds with a development tool used to maintain this project. Every badge is clickable and links back to that project's github / documentation site:

tools / frameworks used by test suite (i.e. used by `make test`):

linters used to maintain code quality (i.e. used by ``make lint``):

tools / frameworks used to render documentation (i.e used by ``make build-docs``):

miscellaneous tools used to maintain this project:

3.3 New Releases

This section serves as a reminder to the maintainers of this project on how to release a new version of this package to PyPI.

Make sure all your changes are committed, that you have added a new section to the [CHANGELOG.md](#) file, and that you have [bumpversion](#) installed. Then run:

```
bumpversion patch # possible values: major / minor / patch
git push
git push --tags
```

A new version of proctor will then deploy to PyPI if all CI checks pass.

PROCTOR

process doctor (PROCTOR)... Utilities related to process / thread management.

project status badges:

version badges:

4.1 Installation

To install `proctor` using `pip`, run the following commands in your terminal:

```
python3 -m pip install --user bolton-proctor # install proctor
```

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

4.2 Useful Links

- [API Reference](#): A developer's reference of the API exposed by this project.
- [cc-python](#): The `cookiecutter` that was used to generate this project. Changes made to this cookiecutter are periodically synced with this project using `cruft`.
- [CHANGELOG.md](#): We use this file to document all notable changes made to this project.
- [CONTRIBUTING.md](#): This document contains guidelines for developers interested in contributing to this project.
- [Create a New Issue](#): Create a new GitHub issue for this project.
- [Documentation](#): This project's full documentation.

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

p

proctor, 3

INDEX

C

`command_exists()` (*in module proctor*), 3

M

module

`proctor`, 3

P

`Process` (*class in proctor*), 3

`proctor`

 module, 3

S

`safe_popen()` (*in module proctor*), 4

T

`to_error()` (*Process method*), 3

U

`unsafe_popen()` (*in module proctor*), 4