
git-toggle

Release 0.1.1

Toggle Git remotes between https and ssh.

Dominic Davis-Foster

Apr 24, 2024

Contents

1	Installation	1
1.1	from PyPI	1
1.2	from GitHub	1
1.3	API Reference	1
1.4	Contributing	3
1.5	Downloading source code	4
1.6	License	5
	Python Module Index	7
	Index	9

Installation

1.1 from PyPI

```
$ python3 -m pip install git-toggle --user
```

1.2 from GitHub

```
$ python3 -m pip install git+https://github.com/domdfcoding/git-toggle@master --user
```

1.3 API Reference

Toggle Git remotes between https and ssh.

Classes:

<code>Remote</code> (style, domain, repo, username)	Represents a remote repository.
<code>Toggle</code> (repo, *args, **kwargs)	Toggle Git remotes between https and ssh.

class Remote (style, domain, repo, username)

Bases: `object`

Represents a remote repository.

Parameters

- **style** (`str`) – The style of remote.
- **domain** (`str`) – The domain of the remote.
- **repo** (`str`) – The repository the remote points to.
- **username** (`str`) – The account on the remote server which owns the repository.

Methods:

<code>as_url()</code>	Returns the <code>apeye.url.URL</code> representation of the <code>:class:`~.Remote`</code> .
<code>from_url(url)</code>	Construct a <code>Remote</code> from a url.
<code>set_repo(repo)</code>	If <code>repo</code> is not <code>None</code> , set <code>repo</code> to that value.
<code>set_username(username)</code>	If <code>username</code> is not <code>None</code> , set <code>username</code> to that value.

Attributes:

<i>domain</i>	The domain of the remote.
<i>repo</i>	The repository the remote points to.
<i>style</i>	The style of remote.
<i>username</i>	The account on the remote server which owns the repository.

as_url()

Returns the `apeye.url.URL` representation of the `:class:`~.Remote``.

Return type `URL`

Returns

domain

Type: `str`

The domain of the remote.

classmethod from_url(url)

Construct a `Remote` from a url.

Parameters `url` (`Union[str, URL]`)

Return type `Remote`

repo

Type: `str`

The repository the remote points to.

set_repo(repo)

If `repo` is not `None`, set `repo` to that value.

Parameters `repo` (`Optional[str]`)

Returns The new value of `repo`

set_username(username)

If `username` is not `None`, set `username` to that value.

Parameters `username` (`Optional[str]`)

Returns The new value of `username`

style

Type: `Literal['https', 'ssh']`

The style of remote.

username

Type: `str`

The account on the remote server which owns the repository.

class Toggler(repo, *args, **kwargs)

Bases: `Repo`

Toggle Git remotes between https and ssh.

Parameters `repo` (`Union[Repo, str, Path, PathLike]`) – The repository to toggle remotes for.

Methods:

<code>get_current_remote([name])</code>	Return the current remote.
<code>set_current_remote(url[, name])</code>	Set the current remote.

get_current_remote (`name='origin'`)

Return the current remote.

Parameters `name` (`str`) – If given, try to retrieve the remote with that name. If no such remote exists returns the `origin` remote. Default `'origin'`.

If no remote can be found an empty string will be returned.

Return type `str`

set_current_remote (`url, name='origin'`)

Set the current remote.

Parameters

- `url` (`Union[str, URL, Remote]`) – The URL to set for the remote.
- `name` (`str`) – The name of the remote to set. Default `'origin'`.

1.4 Contributing

git-toggle uses `tox` to automate testing and packaging, and `pre-commit` to maintain code quality.

Install `pre-commit` with `pip` and install the git hook:

```
$ python -m pip install pre-commit
$ pre-commit install
```

1.4.1 Coding style

`formate` is used for code formatting.

It can be run manually via `pre-commit`:

```
$ pre-commit run formate -a
```

Or, to run the complete autoformatting suite:

```
$ pre-commit run -a
```

1.4.2 Automated tests

Tests are run with `tox` and `pytest`. To run tests for a specific Python version, such as Python 3.6:

```
$ tox -e py36
```

To run tests for all Python versions, simply run:

```
$ tox
```

1.4.3 Type Annotations

Type annotations are checked using `mypy`. Run `mypy` using `tox`:

```
$ tox -e mypy
```

1.4.4 Build documentation locally

The documentation is powered by Sphinx. A local copy of the documentation can be built with `tox`:

```
$ tox -e docs
```

1.5 Downloading source code

The `git-toggle` source code is available on GitHub, and can be accessed from the following URL: <https://github.com/domdfcoding/git-toggle>

If you have `git` installed, you can clone the repository with the following command:

```
$ git clone https://github.com/domdfcoding/git-toggle
```

```
Cloning into 'git-toggle'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 173 (delta 16), reused 17 (delta 6), pack-reused 126
Receiving objects: 100% (173/173), 126.56 KiB | 678.00 KiB/s, done.
Resolving deltas: 100% (66/66), done.
```

Alternatively, the code can be downloaded in a ‘zip’ file by clicking:

Clone or download -> Download Zip

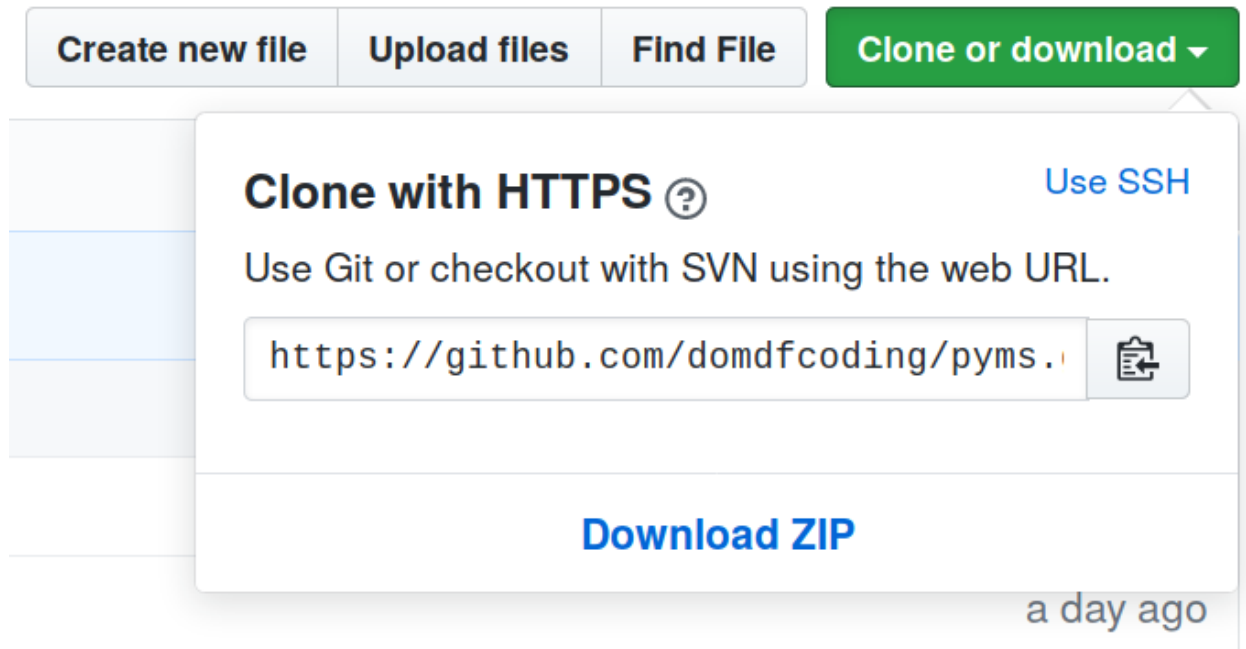


Fig. 1: Downloading a ‘zip’ file of the source code

1.5.1 Building from source

The recommended way to build `git-toggle` is to use `tox`:

```
$ tox -e build
```

The source and wheel distributions will be in the directory `dist`.

If you wish, you may also use `pep517.build` or another **PEP 517**-compatible build tool.

1.6 License

`git-toggle` is licensed under the [MIT License](#)

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions

- Commercial use – The licensed material and derivatives may be used for commercial purposes.
- Modification – The licensed material may be modified.
- Distribution – The licensed material may be distributed.
- Private use – The licensed material may be used and modified in private.

Conditions

- License and copyright notice – A copy of the license and copyright notice must be included with the licensed material.

Limitations

- Liability – This license includes a limitation of liability.
- Warranty – This license explicitly states that it does NOT provide any warranty.

[See more information on choosealicense.com](https://choosealicense.com) ⇒

Copyright (c) 2020 Dominic Davis-Foster

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Python Module Index

g

`git_toggle`, [1](#)

A

`as_url()` (*Remote method*), 2

D

`domain` (*Remote attribute*), 2

F

`from_url()` (*Remote class method*), 2

G

`get_current_remote()` (*Toggler method*), 3

`git_toggle`
module, 1

M

MIT License, 5, 6

module
 `git_toggle`, 1

P

Python Enhancement Proposals
 PEP 517, 5

R

`Remote` (*class in git_toggle*), 1

`repo` (*Remote attribute*), 2

S

`set_current_remote()` (*Toggler method*), 3

`set_repo()` (*Remote method*), 2

`set_username()` (*Remote method*), 2

`style` (*Remote attribute*), 2

T

`Toggler` (*class in git_toggle*), 2

U

`username` (*Remote attribute*), 2