
tuulbachs

Apr 21, 2020

Contents:

1	tuulbash	3
2	tuuldevops	5
2.1	tag_current_version.py	5
3	tuulgit	7
3.1	check_clean.py	7
3.2	tag_commit.py	7
4	tuulpy	9
5	tuulver	11
5.1	version.py	11
6	tuulyaml	13
6.1	parse.py	13
6.2	update_simple_value.py	13
7	Indices and tables	15
	Python Module Index	17
	Index	19

Low-level software tuuls, organized into drawers.

CHAPTER 1

tuulbash

Automation tools for common tasks around software development

2.1 tag_current_version.py

Git tag the commit on the current branch with the current version of this software

```
tuuldevops.tag_current_version.tag_product_version(conf_filename)
```

Git tag the commit on the current branch with the version of this software given in *conf_filename*

An opinionated set of Git tuuls.

3.1 check_clean.py

Check the Git working tree

```
tuulgit.check_clean.is_clean_working_tree()
```

Return a boolean indicating whether the Git working tree is clean or not

3.2 tag_commit.py

Git tag the commit on the current branch, *only if* the working tree is clean

```
tuulgit.tag_commit.tag_current(tag)
```

Git tag (annotated) the current commit from a clean working tree

Raises `TuulError` – when the caller attempts to tag an unclean working tree or to use a tag that already exists on the repo

```
tuulgit.tag_commit.tag_delete_local(tag)
```

Delete the named Git tag (local only). This function does *not* delete remote tags

Raises `TuulError` – if the tag delete fails

CHAPTER 4

tuulpy

5.1 version.py

Utility functions for managing a tuulbachs-formatted version YAML file

`tuulver.version.bump_build(filename)`

Bump the “build” portion of the version from the input YAML filename

`tuulver.version.bump_major(filename)`

Bump the major portion of the version from the input YAML filename

`tuulver.version.bump_minor(filename)`

Bump the minor portion of the version from the input YAML filename

`tuulver.version.bump_pre(filename, prebase='pre')`

Bump the “pre” portion of the version from the input YAML filename

`tuulver.version.emit_product_name(filename)`

Return the product name value from the input YAML filename

`tuulver.version.emit_version(filename)`

Return the version value from the input YAML filename

Low level tuuls for interacting with YAML files.

6.1 parse.py

Parse an input YAML file

```
tuulyaml.parse.parse_yaml (filename)  
    Given input path filename, parse YAML file.
```

6.2 update_simple_value.py

Update a simple top-level value in a YAML file

```
tuulyaml.update_simple_value.update_value (inout_path, existing_key, new_value)  
    Update existing_key to new_value in the existing inout_path YAML file.
```


CHAPTER 7

Indices and tables

- `genindex`
- `modindex`

t

- `tuuldevops.tag_current_version`, [5](#)
- `tuulgit.check_clean`, [7](#)
- `tuulgit.tag_commit`, [7](#)
- `tuulver.version`, [11](#)
- `tuulyaml.parse`, [13](#)
- `tuulyaml.update_simple_value`, [13](#)

B

`bump_build()` (*in module `tuulver.version`*), 11
`bump_major()` (*in module `tuulver.version`*), 11
`bump_minor()` (*in module `tuulver.version`*), 11
`bump_pre()` (*in module `tuulver.version`*), 11

E

`emit_product_name()` (*in module `tuulver.version`*),
11
`emit_version()` (*in module `tuulver.version`*), 11

I

`is_clean_working_tree()` (*in module `tuulgit.check_clean`*), 7

P

`parse_yaml()` (*in module `tuulyaml.parse`*), 13

T

`tag_current()` (*in module `tuulgit.tag_commit`*), 7
`tag_delete_local()` (*in module `tuulgit.tag_commit`*), 7
`tag_product_version()` (*in module `tuuldevops.tag_current_version`*), 5
`tuuldevops.tag_current_version(module)`, 5
`tuulgit.check_clean(module)`, 7
`tuulgit.tag_commit(module)`, 7
`tuulver.version(module)`, 11
`tuulyaml.parse(module)`, 13
`tuulyaml.update_simple_value(module)`, 13

U

`update_value()` (*in module `tuulyaml.update_simple_value`*), 13