tuulbachs

May 11, 2020

Contents:

Low-level software tuuls, organized into drawers.

CHAPTER 1

tuulbash

kickpy - A bash script intended to kick a Python script from an environment that doesn't have an estabished Python environment yet. For example:

```
./kickpy.sh example.py
```

CHAPTER 2

tuulcli

Tuuls for command line interface (CLI).

2.1 cli_color.py

Define colors and font styles for use in CLI output

class tuulcli.cli_color.**CliColor**
Contain the list of colors and font styles

CHAPTER 3

tuuldevops

Automation tuuls for common tasks around software development

3.1 pipeline_steps.py

Provide routines for outputting automated pipeline steps consistently

`tuuldevops.pipeline_steps.`**`major_step`**(*title*, *description*)
Output title and description of a major step in the pipeline

3.2 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

3.3 update_version.py

Update the version in the tuulbachs-formatted YAML version file

`tuuldevops.update_version.`**`update_product_version`**(*conf_filename*, *new_ver*)
Write a new_ver as the new value for the 'version' key in the conf_filename

CHAPTER 4

tuulgit

An opinionated set of Git tuuls.

4.1 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

4.2 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_current_signed**(*tag*)
Git tag (signed) 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 5

tuulpy

subpResult.py - Run a command from Python and return the result. Example:: python subpResult.py

CHAPTER 6

tuulver

Parsing tuuls for a tuulbachs-formatted version YAML input file.

6.1 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_patch**(*filename*)
Bump the patch 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

CHAPTER 7

tuulyaml

Low level tuuls for interacting with YAML files.

7.1 parse.py

Parse an input YAML file

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

7.2 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 8

install

Note that tuulbachs is not yet published at PyPi.

1. Set up and activate a Python virtual environment at the top level of this project
2. `cd` to the local `auto` directory
3. `./install_local.sh`

CHAPTER 9

deploy

Guidance about how to deploy updates to tuulbachs itself

1. Follow *install* guidance
2. Decide which type of semantic version upgrade this is (major, minor, patch, etc.)
3. From `tuulver/version.py`, use the appropriate `bump_*` function to update the version string in `version.yaml`
4. Commit the version change to Git
5. From `tuuldevops/tag_current_version.py`, use the `tag_product_version` function to properly tag this release
6. Push the Git update (including tags) to this repo's remotes

CHAPTER 10

internal API

This is code intended for use by tuulbachs itself, not external users.

10.1 exceptions.py

This module contains the set of tuulbachs' exceptions.

exception tuulbachs.exception.**TuulError**(*msg=None*)
The parent exception from which all other Tuulbachs Python exceptions are derived.

CHAPTER 11

Indices and tables

- genindex
- modindex

Python Module Index

t

Index