tuulbachs

Dave Smith

May 25, 2020

CONTENTS:

1 tuulbash 3

2 tuulcli 5

3 tuuldevops 7

4 tuulgit 9

5 tuulpy 11

6 tuulver 13

7 tuulyaml 15

8 install 17

9 deploy 19

10 internal API 21

11 Indices and tables 23

Low-level software tuuls, organized into drawers.

CHAPTER

ONE

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

TWO

TUULCLI

Tuuls for command line interface (CLI).

CHAPTER

THREE

TUULDEVOPS

Automation tuuls for common tasks around software development

CHAPTER

FOUR

TUULGIT

An opinionated set of Git tuuls.

CHAPTER

FIVE

TUULPY

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

CHAPTER

SIX

TUULVER

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

CHAPTER

SEVEN

TUULYAML

Low level tuuls for interacting with YAML files.

CHAPTER

EIGHT

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. `pip install -r requirements.txt`
3. `cd` to the local `auto` directory
4. `./install_local.sh`

CHAPTER

NINE

DEPLOY

Guidance about how to deploy updates to tuulbachs itself

1. Decide which type of semantic version upgrade this is (major, minor, patch, etc.)
2. From `tuulver/version.py`, use the appropriate `bump_*` function to update the version string in `version.yaml`
3. Follow *install* guidance
4. Commit changes 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

TEN

INTERNAL API

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

CHAPTER

ELEVEN

INDICES AND TABLES

- genindex
- modindex