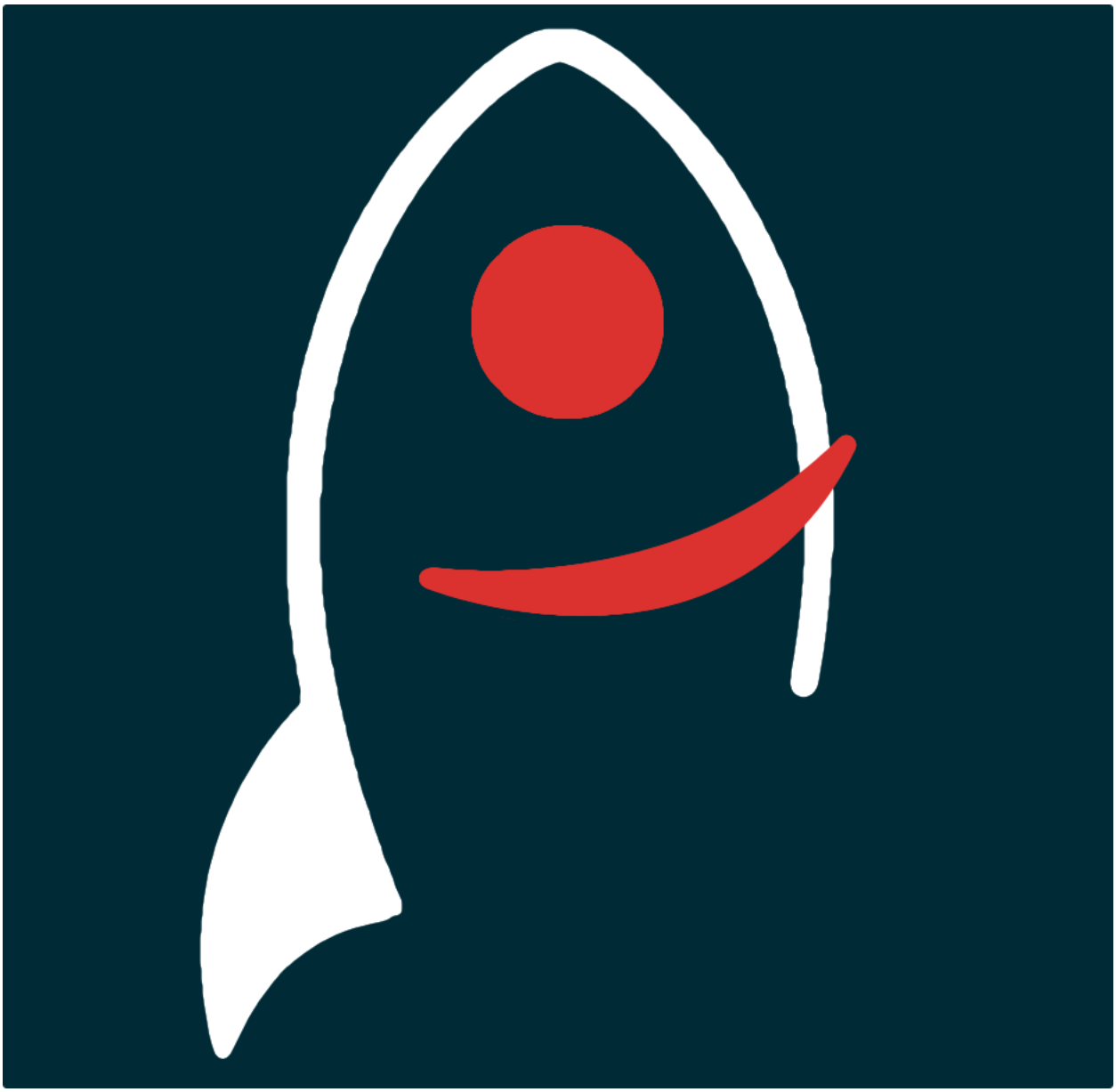

frankenstein Documentation

Release v1.0.2

Dave Young

2023

TABLE OF CONTENTS

1 Features **3**
- 1.1 Installation . . . 3
  - 1.1.1 Development . . . 3
- 1.2 Initialisation . . . 4
  - 1.2.1 Modifying the Settings . . . 4
  - 1.2.2 Basic Python Setup . . . 4
- 1.3 Todo List . . . 4
- 1.4 Release Notes . . . 5

2 API Reference **7**
- 2.1 Modules . . . 7
  - 2.1.1 commonutils *(module)* . . . 7
  - 2.1.2 utKit *(module)* . . . 7
- 2.2 Classes . . . 7
  - 2.2.1 electric *(class)* . . . 7
- 2.3 A-Z Index . . . 8

3 Release Notes **9**

Python Module Index **11**

Index **13**

Project Templates Brought to Life.

Documentation for frankenstein is hosted by Read the Docs (development version and master version). The code lives on github. Please report any issues you find here.

CHAPTER

ONE

FEATURES

-

1.1 Installation

The easiest way to install frankenstein is to use `pip` (here we show the install inside of a conda environment):

```
conda create -n frankenstein python=3.7 pip
conda activate frankenstein
pip install frankenstein
```

Or you can clone the github repo and install from a local version of the code:

```
git clone git@github.com:thespacedoctor/frankenstein.git
cd frankenstein
python setup.py install
```

To upgrade to the latest version of frankenstein use the command:

```
pip install frankenstein --upgrade
```

To check installation was successful run `frankenstein -v`. This should return the version number of the install.

1.1.1 Development

If you want to tinker with the code, then install in development mode. This means you can modify the code from your cloned repo:

```
git clone git@github.com:thespacedoctor/frankenstein.git
cd frankenstein
python setup.py develop
```

Pull requests are welcomed!

1.2 Initialisation

Before using frankenstein you need to use the `init` command to generate a user settings file. Running the following creates a yaml settings file in your home folder under `~/.config/frankenstein/frankenstein.yaml`:

```
frankenstein init
```

The file is initially populated with frankenstein's default settings which can be adjusted to your preference.

If at any point the user settings file becomes corrupted or you just want to start afresh, simply trash the `frankenstein.yaml` file and rerun `frankenstein init`.

1.2.1 Modifying the Settings

Once created, open the settings file in any text editor and make any modifications needed.

1.2.2 Basic Python Setup

If you plan to use `frankenstein` in your own scripts you will first need to parse your settings file and set up logging etc. One quick way to do this is to use the `fundamentals` package to give you a logger, a settings dictionary and a database connection (if connection details given in settings file):

```
## SOME BASIC SETUP FOR LOGGING, SETTINGS ETC
from fundamentals import tools
from os.path import expanduser
home = expanduser("~")
settingsFile  = home + "/.config/frankenstein/frankenstein.yaml"
su = tools(
    arguments={"settingsFile": settingsFile},
    docString=__doc__,
)
arguments, settings, log, dbConn = su.setup()
```

1.3 Todo List

Todo:

- nice!

(The *original entry* is located in /home/docs/checkouts/readthedocs.org/user_builds/frankenstein/checkouts/stable/docs/source/_template line 1.)

1.4 Release Notes

v1.0.2 - May 10, 2022

- **FIXED** doc fixes

v1.0.1 - October 5, 2020

- **FIXED** import issue breaking cl tool

v1.0.0 - October 3, 2020

- Now compatible with Python 3.*

CHAPTER

TWO

API REFERENCE

2.1 Modules

`frankenstein.commonutils`	*common tools used throughout package*
`frankenstein.utKit`	*Unit testing tools*

2.1.1 commonutils *(module)*

common tools used throughout package

2.1.2 utKit *(module)*

Unit testing tools

Classes

`utKit`(moduleDirectory[, dbConn])	*Override dryx utKit*

2.2 Classes

`frankenstein.electric`	*The worker class for the electric module*

2.2.1 electric *(class)*

`class electric`(*log*, *pathToTemplate*, *pathToDestination*, *settings=False*, *ignoreExisting=False*)

Bases: `object`

The worker class for the electric module

Key Arguments

- `log` – logger
- `settings` – the settings dictionary

- `pathToTemplate` – path to the template folder/file
- `pathToDestination` – path to where template should be cloned
- `ignoreExisting` - - ignore existing files in the destination for the template

Methods

`get()`	*do the frankenstein magic!*
`list_placeholders()`	*list the remaining placeholders required by frankenstein*

`get``()`
do the frankenstein magic!

`list_placeholders``()`
list the remaining placeholders required by frankenstein

2.3 A-Z Index

Modules

`frankenstein.commonutils`	*common tools used throughout package*
`frankenstein.utKit`	*Unit testing tools*

Classes

`frankenstein.electric`	*The worker class for the electric module*

Functions

—

CHAPTER

THREE

RELEASE NOTES

v1.0.2 - May 10, 2022

- **FIXED** doc fixes

v1.0.1 - October 5, 2020

- **FIXED** import issue breaking cl tool

v1.0.0 - October 3, 2020

- Now compatible with Python 3.*

PYTHON MODULE INDEX

C

frankenstein.commonutils, 7

U

frankenstein.utKit, 7

INDEX

E

`electric` (*class in frankenstein*), 7

F

`frankenstein.commonutils`
    `module`, 7
`frankenstein.utKit`
    `module`, 7

G

`get()` (*electric method*), 8

L

`list_placeholders()` (*electric method*), 8

M

`module`
    `frankenstein.commonutils`, 7
    `frankenstein.utKit`, 7