



frankenstein Documentation

Release v1.0.2

Dave Young

2023

TABLE OF CONTENTS

1	Features	3
2	How to cite frankenstein	5
2.1	Installation	5
2.1.1	Development	5
2.2	Initialisation	6
2.2.1	Modifying the Settings	6
2.2.2	Basic Python Setup	6
2.3	Todo List	6
2.4	Release Notes	7
3	API Reference	9
3.1	Modules	9
3.1.1	commonutils (<i>module</i>)	9
3.1.2	utKit (<i>module</i>)	9
3.2	Classes	9
3.2.1	electric (<i>class</i>)	9
3.3	A-Z Index	10
4	Release Notes	11
	Python Module Index	13
	Index	15

DOI 10.5281/zenodo.8037492					
downloads	57/month	downloads	57/month	downloads	57/month
downloads	57/month				
coverage	78%	docs	unknown		

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](#).

FEATURES

-

HOW TO CITE FRANKENSTEIN

If you use frankenstein in your work, please cite using the following BibTeX entry:

```
@software{Young_frankenstein,  
  author = {Young, David R.},  
  doi = {10.5281/zenodo.8037492},  
  license = {GPL-3.0-only},  
  title = ,  
  url = {https://github.com/thespacedoctor/frankenstein}  
}
```

2.1 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.

2.1.1 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!

2.2 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`.

2.2.1 Modifying the Settings

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

2.2.2 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()
```

2.3 Todo List

Todo:

- nice!
-

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

2.4 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.*

API REFERENCE

3.1 Modules

<i>frankenstein.commonutils</i>	<i>common tools used throughout package</i>
<i>frankenstein.utKit</i>	<i>Unit testing tools</i>

3.1.1 commonutils (*module*)

common tools used throughout package

3.1.2 utKit (*module*)

Unit testing tools

Classes

<code>utKit(moduleDirectory[, dbConn])</code>	<i>Override dryx utKit</i>
---	----------------------------

3.2 Classes

<i>frankenstein.electric</i>	<i>The worker class for the electric module</i>
------------------------------	---

3.2.1 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

<code>get()</code>	<i>do the frankenstein magic!</i>
<code>list_placeholders()</code>	<i>list the remaining placeholders required by frankenstein</i>

`get()`
do the frankenstein magic!

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

3.3 A-Z Index

Modules

<code>frankenstein.commonutils</code>	<i>common tools used throughout package</i>
<code>frankenstein.utKit</code>	<i>Unit testing tools</i>

Classes

<code>frankenstein.electric</code>	<i>The worker class for the electric module</i>
------------------------------------	---

Functions

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`, 9

U

`frankenstein.utKit`, 9

INDEX

E

`electric` (*class in frankenstein*), 9

F

`frankenstein.commonutils`
module, 9

`frankenstein.utKit`
module, 9

G

`get()` (*electric method*), 10

L

`list_placeholders()` (*electric method*), 10

M

module
 `frankenstein.commonutils`, 9
 `frankenstein.utKit`, 9