



**xxxpython***package\_namexxx* Documentation

***Release v0.1***

**Dave Young**

2022



## TABLE OF CONTENTS

<b>1</b>	<b>Features</b>	<b>3</b>
1.1	Installation . . . . .	3
1.1.1	Development . . . . .	3
1.2	Initialisation . . . . .	4
1.3	Todo List . . . . .	4
1.4	Release Notes . . . . .	5
<b>2</b>	<b>API Reference</b>	<b>7</b>
2.1	Modules . . . . .	7
2.1.1	commonutils ( <i>module</i> ) . . . . .	7
2.1.2	docstring_test ( <i>module</i> ) . . . . .	7
2.1.3	utKit ( <i>module</i> ) . . . . .	11
2.2	Functions . . . . .	12
2.2.1	docsting_test ( <i>function</i> ) . . . . .	13
2.3	A-Z Index . . . . .	16
<b>3</b>	<b>Release Notes</b>	<b>19</b>
	<b>Python Module Index</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



*xxxpackage\_descriptionxxx.*

Documentation for xxxpython\_package\_namexxx is hosted by [Read the Docs](#) ( [development version](#) and [master version](#)). The code lives on [github](#). Please report any issues you find [here](#).



## FEATURES

- 

## 1.1 Installation

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

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

Or you can clone the [github repo](#) and install from a local version of the code:

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

To upgrade to the latest version of `xxxpython_package_namexxx` use the command:

```
pip install xxxpython_package_namexxx --upgrade
```

To check installation was successful run `xxxpython_package_namexxx -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/xxxpython_package_namexxx.git
cd xxxpython_package_namexxx
python setup.py develop
```

[Pull requests](#) are welcomed!

## 1.2 Initialisation

Before using xxxpython<sub>package\_name</sub> 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/xxxpythonpackage_name/xxxpythonpackage_name.yaml`:

```
xxxpythonpackage_name init
```

The file is initially populated with xxxpython<sub>package\_name</sub>'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 xxxpython<sub>package\_name</sub>.yaml file and rerun xxxpython<sub>package\_name</sub> `init`.

## 1.3 Todo List

---

### Todo:

- update key arguments values and definitions with defaults
  - update return values and definitions
  - update usage examples and text
  - update docstring text
  - check sublime snippet exists
  - clip any useful text to docs mindmap
  - regenerate the docs and check redendering of this docstring
- 

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/python-package-frankenstein-template/envs/develop/lib/python3.7/site-packages/xxxpythonpackage_name-0.1-py3.7.egg/xxxpythonpackage_name/docstring_test.py:docstring of xxxpythonpackage_name.docstring_test.docsting_test, line 37.)`

---

### Todo:

- nice!
- 

(The [original entry](#) is located in `/home/docs/checkouts/readthedocs.org/user_builds/python-package-frankenstein-template/envs/develop/lib/python3.7/site-packages/xxxpythonpackage_name-0.1-py3.7.egg/xxxpythonpackage_name/docstring_test.py:docstring of xxxpythonpackage_name.docstring_test.docsting_test, line 53.)`

---

### Todo:

- update key arguments values and definitions with defaults
- update return values and definitions
- update usage examples and text
- update docstring text



- check sublime snippet exists
  - clip any useful text to docs mindmap
  - regenerate the docs and check redendering of this docstring
- 

(The [original entry](#) is located in /home/docs/checkouts/readthedocs.org/user\_builds/python-package-frankenstein-template/envs/develop/lib/python3.7/site-packages/xxxpython\_package\_namexxx-0.1-py3.7.egg/xxxpython\_package\_namexxx/docstring\_test.py:docstring of xxxpython\_package\_namexxx.docstring\_test.docsting\_test, line 37.)

---

**Todo:**

- nice!
- 

(The [original entry](#) is located in /home/docs/checkouts/readthedocs.org/user\_builds/python-package-frankenstein-template/envs/develop/lib/python3.7/site-packages/xxxpython\_package\_namexxx-0.1-py3.7.egg/xxxpython\_package\_namexxx/docstring\_test.py:docstring of xxxpython\_package\_namexxx.docstring\_test.docsting\_test, line 53.)

---

**Todo:**

- Make sure todo list is working. ✓
- 

(The *original entry* is located in /home/docs/checkouts/readthedocs.org/user\_builds/python-package-frankenstein-template/checkouts/develop/docs/source/\_template\_.md, line 1.)

## 1.4 Release Notes



## API REFERENCE

### 2.1 Modules

<code>xxxpython_package_namexxx.commonutils</code>	<i>common tools used throughout package</i>
<code>xxxpython_package_namexxx.docstring_test</code>	<i>Testing the rendering of docstings - feel free to remove this module</i>
<code>xxxpython_package_namexxx.utKit</code>	<i>Unit testing tools</i>

#### 2.1.1 commonutils (module)

*common tools used throughout package*

##### Sub-modules

---

#### 2.1.2 docstring\_test (module)

*Testing the rendering of docstings - feel free to remove this module*

Check the API output here: [file:///Users/Dave/git\\_repos/templates/python-package-template/xxxpython\\_package\\_namexxx/docs/build/html/\\_api/xxxpython\\_package\\_namexxx.docstring\\_test.html](file:///Users/Dave/git_repos/templates/python-package-template/xxxpython_package_namexxx/docs/build/html/_api/xxxpython_package_namexxx.docstring_test.html)

##### Functions

<code>docstring_test(self)</code>	<i>the database object for sherlock, setting up ssh tunnels and various database connections</i>
-----------------------------------	--

## Sub-modules

---

### `docstring_test` (*self*)

*the database object for sherlock, setting up ssh tunnels and various database connections*

The returned dictionary of database connections contain the following databases:

- `transients` – the database hosting the transient source data
- `catalogues` – connection to the database hosting the contextual catalogues the transients are to be crossmatched against
- `marshall` – connection to the PESSTO Marshall database

#### Key Arguments:

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

#### Return:

- `dbConns` – a dictionary of the database connections required by sherlock

#### Usage:

To setup the sherlock database connections, run the following:

```
.. code-block:: python

    # SETUP ALL DATABASE CONNECTIONS
    from sherlock import database
    db = database(
        log=log,
        settings=settings
    )
    dbConns, dbVersions = db.connect()
    transientsDbConn = dbConns["transients"]
    cataloguesDbConn = dbConns["catalogues"]
    pmDbConn = dbConns["marshall"]
```

---

#### Todo:

- update key arguments values and definitions with defaults
  - update return values and definitions
  - update usage examples and text
  - update docstring text
  - check sublime snippet exists
  - clip any useful text to docs mindmap
  - regenerate the docs and check rendering of this docstring
- 

#### Embed reStructuredText

**Todo:**

- nice!

**Code and Syntax Highlighting**

Inline code has back-ticks around it.

```
var s = "JavaScript syntax highlighting";
alert(s);
```

```
s = "Python syntax highlighting"
print s
myString = "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
↳ eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
↳ veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
↳ consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
↳ cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
↳ proident, sunt in culpa qui officia deserunt mollit anim id est laborum."
```

No language indicated, so no syntax highlighting.  
But let's throw in a `<b>tag</b>`.

**Mermaid**

```
gantt
    dateFormat YYYY-MM-DD
    title Adding GANTT diagram functionality to mermaid
    section A section
    Completed task                :done,    des1, 2014-01-06,2014-01-08
    Active task                    :active,  des2, 2014-01-09, 3d
    Future task                    :         des3, after des2, 5d
    Future task2                   :         des4, after des3, 5d
    section Critical tasks
    Completed task in the critical line :crit, done, 2014-01-06,24h
    Implement parser and jison        :crit, done, after des1, 2d
    Create tests for parser           :crit, active, 3d
    Future task in critical line      :crit, 5d
    Create tests for renderer         :2d
    Add to mermaid                   :1d
```

**Flowchart**

```
s=>start: start
e=>end: end
o=>operation: operation
sr=>subroutine: subroutine
c=>condition: condition
i=>inputoutput: inputoutput
p=>parallel: parallel

s->o->c
c(yes)->i->e
c(no)->p
p(path1, bottom)->sr(right)->o
p(path2, top)->o
```

## Tables

Colons can be used to align columns.

Tables	Are	Cool
col 3 is	right-aligned	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

## Definitions

**term** definition

**what** Definition lists associate a term with a definition.

## Math

A formula,  $e^{i\pi} + 1 = 0$ , inside a paragraph.

$$e^{i\pi} + 1 = 0$$

## Super/Sub Scripts

$m^2$

$x^{2,y}$

$x_z$

$C_6H_{12}O_6$

## Citations

Cite a source.[p. 42]<sup>1</sup>

Black (2015)<sup>2</sup>

## Task Lists

- [x] Completed task item
- [ ] Unfinished task item

## Footnote

Here's a sentence with a footnote<sup>3</sup> in the middle of it!

## Emphasis

Emphasis, aka italics, with *asterisks*

Strong emphasis, aka bold, with **asterisks**.

Strikethrough .

## Lists

1. First ordered list item
2. Another item
  - Unordered sub-list.
3. Actual numbers don't matter, just that it's a number

---

<sup>1</sup> John Doe. *A Totally Fake Book*. Vanity Press, 2006.

<sup>2</sup> A Bryden Black 2015, *The Lion, the Dove, & the Lamb*, Wipf and Stock Publishers

<sup>3</sup> This is the footnote.

1. Ordered sub-list
4. And another item.

You can have properly indented paragraphs within list items.

and unordered lists:

- Unordered list can use asterisks
- Or minuses
- Or pluses

### Links

I'm an inline-style link

### Images

Here's our logo (hover to see the title text):

Inline-style:



Reference-style:



### Blockquotes

Blockquotes are very handy in email to emulate reply text. This line is part of the same quote.

Quote break.

This is a very long line that will still be quoted properly when it wraps. Oh boy let's keep writing to make sure this is long enough to actually wrap for everyone. Oh, you can *put* **Markdown** into a blockquote.

### Abbreviations

The HTML specification is maintained by the W3C.

[HTML]: Hyper Text Markup Language[W3C]: World Wide Web Consortium

### Horizontal Rule

Three or more...

Hyphens

## 2.1.3 utKit (module)

*Unit testing tools*

## Classes

---

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

---

## Sub-modules

---

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

---

**class** `utKit` (*moduleDirectory*, *dbConn=False*)

Bases: `fundamentals.utKit.utKit`

*Override dryx utKit*

**get\_project\_root** ()

*Get the root of the ``python`` package - useful for getting files in the root directory of a project*

**Return:**

- `rootPath` – the root path of a project

**refresh\_database** ()

*Refresh the unit test database*

**setupModule** ()

*The setupModule method*

**Return**

- `log` – a logger
- `dbConn` – a database connection to a test database (details from yaml settings file)
- `pathToInputDir` – path to modules own test input directory
- `pathToOutputDir` – path to modules own test output directory

**tearDownModule** ()

*The tearDownModule method*

## 2.2 Functions

---

<code>xxxpython_package_namexxx.docstring_test.docstring_test</code>	<i>the database object for sherlock, setting up ssh tunnels and various database connections</i>
--	--

---



### 2.2.1 docsting\_test (function)

**docsting\_test** (*self*)

*the database object for sherlock, setting up ssh tunnels and various database connections*

The returned dictionary of database connections contain the following databases:

- `transients` – the database hosting the transient source data
- `catalogues` – connection to the database hosting the contextual catalogues the transients are to be crossmatched against
- `marshall` – connection to the PESSTO Marshall database

**Key Arguments:**

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

**Return:**

- `dbConns` – a dictionary of the database connections required by sherlock

**Usage:**

To setup the sherlock database connections, run the following:

```
.. code-block:: python

    # SETUP ALL DATABASE CONNECTIONS
    from sherlock import database
    db = database(
        log=log,
        settings=settings
    )
    dbConns, dbVersions = db.connect()
    transientsDbConn = dbConns["transients"]
    cataloguesDbConn = dbConns["catalogues"]
    pmDbConn = dbConns["marshall"]
```

---

**Todo:**

- update key arguments values and definitions with defaults
  - update return values and definitions
  - update usage examples and text
  - update docstring text
  - check sublime snippet exists
  - clip any useful text to docs mindmap
  - regenerate the docs and check rendering of this docstring
- 

**Embed reStructuredText**

---

**Todo:**

- nice!

## Code and Syntax Highlighting

Inline code has back-ticks around it.

```
var s = "JavaScript syntax highlighting";  
alert(s);
```

```
s = "Python syntax highlighting"  
print s  
myString = "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do  
↪eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim  
↪veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo  
↪consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse  
↪cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non  
↪proident, sunt in culpa qui officia deserunt mollit anim id est laborum."
```

No language indicated, so no syntax highlighting.  
But let's throw in a `<b>tag</b>`.

## Mermaid

```
gantt  
    dateFormat YYYY-MM-DD  
    title Adding GANTT diagram functionality to mermaid  
    section A section  
        Completed task           :done,    des1, 2014-01-06,2014-01-08  
        Active task               :active,   des2, 2014-01-09, 3d  
        Future task               :         des3, after des2, 5d  
        Future task2              :         des4, after des3, 5d  
    section Critical tasks  
        Completed task in the critical line :crit, done, 2014-01-06,24h  
        Implement parser and jison          :crit, done, after des1, 2d  
        Create tests for parser             :crit, active, 3d  
        Future task in critical line        :crit, 5d  
        Create tests for renderer           :2d  
        Add to mermaid                     :1d
```

## Flowchart

```
s=>start: start  
e=>end: end  
o=>operation: operation  
sr=>subroutine: subroutine  
c=>condition: condition  
i=>inputoutput: inputoutput  
p=>parallel: parallel  
  
s->o->c  
c(yes)->i->e  
c(no)->p  
p(path1, bottom)->sr(right)->o  
p(path2, top)->o
```

## Tables

Colons can be used to align columns.

Tables	Are	Cool
col 3 is	right-aligned	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

## Definitions

**term** definition

**what** Definition lists associate a term with a definition.

## Math

A formula,  $e^{i\pi} + 1 = 0$ , inside a paragraph.

$\{e\}^{i\pi} + 1 = 0$

## Super/Sub Scripts

$m^2$

$x^{2,y}$

$x_z$

$C_6H_{12}O_6$

## Citations

Cite a source. [p. 42]<sup>1</sup>

Black (2015)<sup>2</sup>

## Task Lists

- [x] Completed task item
- [ ] Unfinished task item

## Footnote

Here's a sentence with a footnote<sup>3</sup> in the middle of it!

## Emphasis

Emphasis, aka italics, with *asterisks*

Strong emphasis, aka bold, with **asterisks**.

Strikethrough .

## Lists

1. First ordered list item
2. Another item
  - Unordered sub-list.
3. Actual numbers don't matter, just that it's a number
  1. Ordered sub-list
4. And another item.

---

<sup>1</sup> John Doe. *A Totally Fake Book*. Vanity Press, 2006.

<sup>2</sup> A Bryden Black 2015, *The Lion, the Dove, & the Lamb*, Wipf and Stock Publishers

<sup>3</sup> This is the footnote.

You can have properly indented paragraphs within list items.

and unordered lists:

- Unordered list can use asterisks
- Or minuses
- Or pluses

### Links

I'm an inline-style link

### Images

Here's our logo (hover to see the title text):

Inline-style:



Reference-style:



### Blockquotes

Blockquotes are very handy in email to emulate reply text. This line is part of the same quote.

Quote break.

This is a very long line that will still be quoted properly when it wraps. Oh boy let's keep writing to make sure this is long enough to actually wrap for everyone. Oh, you can *put* **Markdown** into a blockquote.

### Abbreviations

The HTML specification is maintained by the W3C.

[HTML]: Hyper Text Markup Language[W3C]: World Wide Web Consortium

### Horizontal Rule

Three or more...

Hyphens

## 2.3 A-Z Index

### Modules

<code>xxxpython_package_namexxx.commonutils</code>	<i>common tools used throughout package</i>
<code>xxxpython_package_namexxx.docstring_test</code>	<i>Testing the rendering of docstings - feel free to remove this module</i>
<code>xxxpython_package_namexxx.utKit</code>	<i>Unit testing tools</i>

### Classes

### Functions

---

xxxpython_package_namexxx. docstring_test.docsting_test	<i>the database object for sherlock, setting up ssh tunnels and various database connections</i>
--	--

---



**RELEASE NOTES**





## PYTHON MODULE INDEX

### C

`xxxpython_package_namexxx.commonutils,`  
[7](#)

### d

`xxxpython_package_namexxx.docstring_test,`  
[7](#)

### u

`xxxpython_package_namexxx.utKit,` [11](#)



## INDEX

### D

`docstring_test()` (in *module xxxpython\_package\_namexxx.docstring\_test*),  
8, 13

### G

`get_project_root()` (*utKit method*), 12

### M

*module*  
    `xxxpython_package_namexxx.commonutils`,  
    7  
    `xxxpython_package_namexxx.docstring_test`,  
    7  
    `xxxpython_package_namexxx.utKit`, 11

### R

`refresh_database()` (*utKit method*), 12

### S

`setupModule()` (*utKit method*), 12

### T

`tearDownModule()` (*utKit method*), 12

### U

`utKit` (*class in xxxpython\_package\_namexxx.utKit*), 12

### X

`xxxpython_package_namexxx.commonutils`  
    *module*, 7  
`xxxpython_package_namexxx.docstring_test`  
    *module*, 7  
`xxxpython_package_namexxx.utKit`  
    *module*, 11