
comb Documentation

Release 0.9.10

WANG WENPEI

January 14, 2016

1	Quick links	3
2	Documents	5
2.1	Guide	5
2.2	Code view	6
2.3	Release notes	8
	Python Module Index	11

`comb` , A simple and high-performance framework for create parallel program

Quick links

- [Documentation](#)
- [Source \(github\)](#)

2.1 Guide

2.1.1 Installation

Automatic installation:

```
pip install comb
```

comb is listed in [PyPI](#) and can be installed with `pip` or `easy_install`. it includes demo applications.

Manual installation: Download the latest source from [Github](#).

```
git clone https://github.com/kbonez/comb.git
cd comb
python setup.py build
sudo python setup.py install
```

The comb source code is [hosted on GitHub](#).

Prerequisites: comb was only test on Python 2.7. It may be runs on all Python versions.

2.1.2 How to use

To use comb, you should create a python module file. we named *slot*.

A legal slot must be named 'Slot' in your module file and it must be at least contain four method:

- initialize
 initial resource, e.g: database handle
- `__enter__`
 get next data to do,you can fetch one or more data.
- slot
 user custom code
- `__exit__`
 when slot finished, call this method

2.1.3 Slot Demo

- Deal List Data
- Deal Mongo MQ
- Recycle Mongo MQ
- Custom Slot options

2.1.4 Start

- Execute a comb is very simple. just execute:

```
comb --root SLOT_ROOT_PATH slot-package.slot-module
```

if you set **SLOTPATH** environment, you can use:

```
comb slot-package.slot-module
```

- Quick View. call:

```
comb comb.demo.list
```

this will execute the [Deal List Demo](#)

2.1.5 Note

- You'd better make comb process number equals your cpu core number.
- We strongly recommend you use comb with [supervisor](#)

2.2 Code view

2.2.1 `comb.slot` — Slot base class

Slot

class `comb.slot.Slot` (*combd*)

To use comb, you should create a python module file. we named *slot*.

A legal slot must be named 'Slot' in your module file and it must be at least contain four method:

- *initialize*

initial resource, e.g: database handle

- *__enter__*

get next data to do,you can fetch one or more data.

- *slot*

user custom code

- *__exit__*

when slot finished, call this method

Don't override this method unless what you're doing.

`Slot.__init__(combd)`

Don't override this method unless what you're doing.

`Slot.__enter__()`

You **MUST** return False when no data to do.

The return value will be used in `Slot.slot`

`Slot.__exit__(exc_type, exc_val, exc_tb)`

When slot done, will call this method.

`Slot.initialize()`

Hook for subclass initialization.

This block is execute before thread initial

Example:

```
class UserSlot(Slot):
    def initialize(self):
        self.threads_num = 10

    def slot(self, result):
        ...
```

`Slot.slot(msg)`

Add your custom code at here.

For example, look at:

- `comb.demo.list`
- `comb.demo.mongo`
- `comb.demo.redis`

2.2.2 `comb.demo.*` — Slot demo classes

List.Slot

`class comb.demo.list.Slot(combd)`

Don't override this method unless what you're doing.

`Slot.initialize()`

Hook for initialization.

This block is execute before thread initial

Example:

```
class UserSlot(Slot):
    def initialize(self):
        ...

    def slot(self, result):
        ...
```

`Slot.__enter__()`

Slot.**slot** (*result*)

Slot.**__exit__** (*exc_type, exc_val, exc_tb*)

Redis.Slot

class comb.demo.redis.**Slot** (*combd*)

Don't override this method unless what you're doing.

Slot.**initialize** ()

This block is execute before thread initial

Example:

```
class UserSlot(Slot):
    def initialize(self, *args, **kwargs):
        self.attr = kwargs.get('attr', None)

    def slot(self, result):
        ...
```

Slot.**__enter__** ()

Slot.**slot** (*result*)

Slot.**__exit__** (*exc_type, exc_val, exc_tb*)

2.3 Release notes

2.3.1 Planning feature

- Comb cook book, chinese version
- move domain kbonez.com to leadyi.com

2.3.2 What's new in Comb 0.9.x

- **Version 0.9.6**
 - Add `--no-daemon` option.
- **Version 0.9.5**
 - User interrupt directly when user specific `--debug` option.
- **Version 0.9.4**
 - change `--sleep` data type to float in python3.
- **Version 0.9.3**
 - update document
 - improve setup.py file
- **Version 0.9.2**
 - user interrupt more frequently.
- **Version 0.9.1**

- support python3
- **Version 0.9.0**
 - Add `--once` option. execute once and exit instead of wait.
 - As of Comb 0.8.99, we change `--sleep_max` option to `--sleep-max`.
 - change comb script install path,use python library path instead `/usr/local/bin`
- `genindex`
- `modindex`

C

`comb.demo.list`, 7
`comb.demo.redis`, 8
`comb.slot`, 6

Symbols

`__enter__()` (comb.demo.list.Slot method), 7
`__enter__()` (comb.demo.redis.Slot method), 8
`__enter__()` (comb.slot.Slot method), 7
`__exit__()` (comb.demo.list.Slot method), 8
`__exit__()` (comb.demo.redis.Slot method), 8
`__exit__()` (comb.slot.Slot method), 7
`__init__()` (comb.slot.Slot method), 7

C

comb.demo.list (module), 7
comb.demo.redis (module), 8
comb.slot (module), 6

I

`initialize()` (comb.demo.list.Slot method), 7
`initialize()` (comb.demo.redis.Slot method), 8
`initialize()` (comb.slot.Slot method), 7

S

Slot (class in comb.demo.list), 7
Slot (class in comb.demo.redis), 8
Slot (class in comb.slot), 6
`slot()` (comb.demo.list.Slot method), 7
`slot()` (comb.demo.redis.Slot method), 8
`slot()` (comb.slot.Slot method), 7