
Cabrita Documentation

Release 2.0.0

Chris Maillefaud

Jun 16, 2018

Contents:

1	Cabrita API Reference	3
1.1	cabrita.command	3
1.2	cabrita.versions	4
1.3	cabrita.abc	4
1.4	cabrita.abc.base	4
1.5	cabrita.abc.utils	5
1.6	cabrita.components	6
1.7	cabrita.components.box	7
1.8	cabrita.components.config	9
1.9	cabrita.components.dashboard	11
1.10	cabrita.components.docker	12
1.11	cabrita.components.git	13
1.12	cabrita.components.watchers	14
2	Indices and tables	17
	Python Module Index	19

1.1 cabrita.command

Command module.

This module has the CabritaCommand class, which is responsible for:

1. Load and check for valid cabrita.yml file
2. Load and check for valid docker-compose.yml files
3. Generate and add docker services to boxes
4. Generate and add watchers to dashboard

```
class cabrita.command.CabritaCommand(cabrita_path: str, compose_path: tuple, back-  
ground_color: Union[str, NoneType] = 'black', ver-  
sion: str = 'dev') → None
```

Bases: object

Cabrita Command class.

background_color

Return Background Color enum.

Returns BoxColor instance

execute() → None

Execute dashboard to show data in terminal.

Returns None

has_a_valid_compose

Return if Compose data is valid.

Returns bool

has_a_valid_config

Return if Config data is valid.

Returns bool

prepare_dashboard() → None
Prepare the dashboard.

Returns None

read_compose_files() → None
Read docker compose files data.

Returns None

1.2 cabrita.versions

Version package.

Checks version number for upgrades in PyPI

cabrita.versions.check_version() → str
Check if it is the latest version.

Compares actual version vs last known version in PyPI, for upgrades

Returns string

cabrita.versions.versions() → Union[List[str], NoneType]
Return the version list data from PyPI.

Returns list

1.3 cabrita.abc

Cabrita base package.

Contains abstract classes and general purpose code.

1.4 cabrita.abc.base

Base Module.

1.4.1 ConfigTemplate

Base class for the config template object. This template will process the YAML files. Subclasses are: Config and Compose classes

1.4.2 InspectTemplate

Base class for the Inspector template object. This template will process docker and git status for compose services
Subclasses are: DockerInspect and GitInspect

class cabrita.abc.base.ConfigTemplate → None
Bases: abc.ABC

Abstract class for processing yaml files.

add_path (*path: str, base_path: str = '/home/docs/checkouts/readthedocs.org/user_builds/cabrita/checkouts/stable/source'*)
 → None
 Add new path for list for each yaml file.

base_path
 Return base path for yaml file.

is_valid
 Check if yaml is valid.

load_file_data () → None
 Load data from yaml file.
 First file will be the main file. Every other file will override data, on reversed order list:

Example:

1. docker-compose.yml (main file)
2. docker-compose-dev.yml (override main)
3. docker-compose-pycharm.yml (override dev)

version
 Return version value inside yaml file.

class cabrita.abc.base.**InspectTemplate** (*compose, interval: int*) → None
 Bases: abc.ABC

Abstract class for compose service inspectors.

can_update
 Check if inspector can inspect again.

inspect (*service: str*) → None
 Run inspect code.

status (*service*)
 Return service status.

If can update, start fetching new data calling self.inspect method.

Parameters **service** – docker service name

Returns dict or dashing obj. If not fetch data yet send default widget.

1.5 cabrita.abc.utils

Base utils module.

cabrita.abc.utils.format_color (*text: str, style: str, theme: str = None*) → str
 Format string with color using formatStr method.

cabrita.abc.utils.get_path (*path: str, base_path: str*) → str
 Return real path from string.

Converts environment variables to path Converts relative path to full path

cabrita.abc.utils.get_sentry_client () → Union[raven.base.Client, NoneType]
 Return synchronous Sentry client if DSN is available.

```
cabrita.abc.utils.run_command(task, get_stdout=False)
```

Run subprocess command.

The function will attempt to run the task informed and return a boolean for the exit code or the captured std from console.

Arg task (string): the command to run

Arg get_stdout (bool, optional): capture stdout

Arg run_stdout (bool, optional): capture and run stdout

Returns bool or string

1.6 cabrita.components

Components Sub-Package.

This package has the cabrita components for:

box = the Box class (the building block for the dashboard)

config = the Config class (the data from cabrita.yml file)

dashboard = the Dashboard class (convert boxes to dashing widgets and display it)

docker = the DockerInspect class (the runner for inspect docker containers)

git = the GitInspect class (the runner for inspect git data)

watchers = the Watch class (the collection of internal and user watchers for the dashboard)

```
class cabrita.components.BoxColor
```

Bases: `enum.Enum`

Enum using [Blessing Colors](#) values.

Currently not working correctly.

Color	Normal	Bright
black	0	8
red	1	9
green	2	10
yellow	3	11
blue	4	12
magenta	5	13
cyan	6	14
white	7	15

```
black = 16
```

```
blue = 4
```

```
cyan = 14
```

```
grey = 0
```

```
white = 7
```

```
yellow = 11
```

1.7 cabrita.components.box

Box Module.

This module has the Box Class, which is the building block for dashboards.

Each box updates his data in a separate thread in Python.

```
class cabrita.components.box.Box (background_color:          cabrita.components.BoxColor
                                = <BoxColor.black:          16>,          compose:
                                cabrita.components.config.Compose = None,      git:
                                cabrita.components.git.GitInspect  = None,      docker:
                                cabrita.components.docker.DockerInspect = None) →
                                None
```

Bases: object

Box Class.

The building block of the dashboard.

This class is called inside the CabritaCommand class, using the data from Config and Compose classes.

add_service (service: str) → None

Append new service in box services list.

Parameters service – service name

Returns None

background_color

Return the Box Color Enum.

Returns BoxColor object

can_update

Check if box data can be updated.

Returns bool

categories

Return if box will show services as columns (categories).

Default: Show as lines.

Returns list

static format_revision (table_lines: list) → list

Format revision info to make all lines have the same width.

Example:

```
Revision 1: master@1234abc
Revision 2: epic/refactoring@5678def

After formatting:

Revision 1:          master@1234abc
Revision 2: epic/refactoring@5678def
```

Parameters table_lines – list

Returns list

includes

Return the list of the only services which will be included in this box.

Default is: no filters. This option is mutually exclusive with the 'main' parameter.

Returns list

interval

Return interval in seconds for each box data update.

Minimum: 0.5.

Returns float

load_data (*data: dict*) → None

Add data from config class in box.

Parameters **data** – config data parsed from cabrita.yml file.

Returns None

main

Return if this is the main box (the 'main' box parameter).

Default: No. Only one box can be true.

Returns bool

port_detail

Return if box will show external, internal or both docker container exposed ports.

Works in conjunction with 'port_view' parameter. Default: external ports.

Returns PortDetail enum property

port_view

Return if box will show docker container port info (the 'port_view' box parameter).

Default: hidden

Returns PortView enum property

run () → None

Run main code for update box data.

Updates the box widget property.

Returns None

services

Return unique sorted service names inside box.

Returns list

show_git

Return if box will show git information (the 'show_git' box parameter).

Default: True

Returns bool

show_revision

Return if box will show git tag/commit hash (the 'show_revision' box parameter).

Default: False

Returns bool

size

Return box size to render in dashboard (the 'size' box parameter).

Default: "large".

Returns string

title

Return box title name (the 'name' box parameter).

Default: "Box".

Returns string

widget

Return dashing widget object.

Returns dashing object

`cabrita.components.box.update_box(box)`

Update box data.

This method are called by a thread class to update.

Parameters **box** – the box to update

Returns dashing object

1.8 cabrita.components.config

Config module.

This module has: The **Config** class, which is responsible for handling program options from cabrita.yml file.

The **Compose** class, which is responsible for handling docker-compose data from yamls.

class `cabrita.components.config.Compose` → None

Bases: `cabrita.abc.base.ConfigTemplate`

Main class for Docker-Compose data.

get_build_path (*service_name: str*) → str

Get build full path for service.

Parameters **service_name** – docker service name

Returns str

get_from_service (*service_name: str, key: str*) → Any

Get value from key for informed service.

Example: `get_from_service("flower", "ports")` returns ["5555"]

Parameters

- **service_name** – docker service name
- **key** – search key for service data

Returns List, String, Dict or None

get_ports_from_service (*service: str*) → Union[List, NoneType]

Return ports from service.

The ports can be from the 'ports' or 'expose' parameters in yaml.

Parameters `service` – service name

Returns List or None

is_image (*service_name: str*) → bool

Check if service are built from image or dockerfile.

Parameters `service_name` – docker service name

Returns bool

is_valid

Return if docker-compose are valid.

Returns bool

networks

Return networks configuration in docker-compose yaml files.

Returns dict

services

Return services configuration in docker-compose yaml files.

Returns dict

volumes

Return volumes configuration in docker-compose yaml files.

Returns dict

class `cabrita.components.config.Config` → None

Bases: `cabrita.abc.base.ConfigTemplate`

Cabrita Configuration main class.

background_color

Return background color for box.

Parameter: ‘background_color’. Options: Black, Blue, Cyan, Grey, Yellow, White. Default: Black.

Returns BoxColor instance.

background_color_value

Return blessed box color value from enum.

Returns int

boxes

Return box configuration data.

Parameter: ‘boxes’. No default available.

Returns dict

compose_files

Return docker-compose paths list.

Parameter: ‘compose_files’. Default is empty.

Returns list

generate_boxes (*services: dict*)

Autogenerate boxes in version 0 runs.

The number of boxes are determined by terminal size. The columns visible inside each one are determined by the number of boxes generated.

Parameters `services` – services to be included in dashboard.

Returns dict

static `get_compose_path` (*compose_path: str, base_path: str*) → str

Get docker-compose file full path.

The compose path can be absolute or relative - the ‘base_path’ will be used to resolve full path.

Parameters

- `compose_path` – the path for docker-compose file.
- `base_path` – the base path for cabrita.yml file.

Returns str

ignore_services

Return ignore services in dashboard.

Parameter: ‘ignore_services’. Default: All services viewed.

Returns list

is_valid

Return if configuration is valid.

Calls for the version founded inside yml file: Version 0: No yml available - calls `_check_v0()`. Version 1: Deprecated - calls `_check_v1()`. Version 2: Last version - calls `_check_v2()`.

Returns bool

layout

Return dashboard layout.

Parameter: ‘layout’ Options are: ‘horizontal’ and ‘vertical’. Default: ‘horizontal’.

Returns str

title

Return dashboard title.

Parameter: ‘title’. Default: ‘Docker-Compose’.

Returns str

watchers

Return watchers configuration data.

Parameter: ‘watchers’.

No default available. :return:

1.9 cabrita.components.dashboard

Dashboard module.

This module contains the Dashboard class, which is responsible to build all dashing widgets from boxes generate the layout and display it in terminal.

class `cabrita.components.dashboard.Dashboard` (*config: cabrita.components.config.Config*)

Bases: object → None

Dashboard class.

add_box (*box: cabrita.components.box.Box*) → None
Add new box to dashboard.

Parameters **box** – Box to be added

Returns None

all_boxes
Return all boxes widgets in order.

Returns list

run () → None
Run dashboard code.

This code starts fullscreen mode, hides cursor and display the generated layout. To stop press 'q' or 'ctrl-c'.

Returns None

1.10 cabrita.components.docker

Docker module.

This module contains the DockerInspect class which is responsible to inspect docker data from each service in dashboard.

class cabrita.components.docker.DockerInspect (*compose: cabrita.components.config.Compose,*
interval: int, port_view:
cabrita.components.docker.PortView,
port_detail:
cabrita.components.docker.PortDetail,
files_to_watch: List[str], ser-
vices_to_check_git: List[str]) → None

Bases: *cabrita.abc.base.InspectTemplate*

DockerInspect class.

inspect (*service: str*) → None
Inspect docker container.

Parameters **service** – service name as defined in docker-compose.yml.

Returns None

class cabrita.components.docker.PortDetail
Bases: *enum.Enum*

Port Detail for docker ports.

Ports are determined by the 'ports' and 'expose' parameters inside docker-compose.yml files.

- **external**: show external ports only
- **internal**: show internal ports only
- **both**: show both ports.

both = 'both'

external = 'external'

internal = 'internal'

```
class cabrita.components.docker.PortView
```

Bases: `enum.Enum`

Port View for docker ports.

Defines where to show port information on box.

Options:

- **hidden:** Do not show info.
- **column:** Show ports info in a separate column
- **name:** Show ports info after service name
- **status:** Show ports info after service status

```
column = 'column'
```

```
hidden = 'hidden'
```

```
status = 'status'
```

1.11 cabrita.components.git

Git module.

This module has the `GitInspect` class, which is responsible for inspect git data from docker services git branches in dashboard.

```
class cabrita.components.git.GitDirection
```

Bases: `enum.Enum`

Git direction for branch evaluation.

Options:

- **ahead:** check for commits ahead branch
- **behind:** check for commits behind branch

```
ahead = 1
```

```
behind = 2
```

```
class cabrita.components.git.GitInspect (compose: cabrita.components.config.Compose, interval: int, target_branch: str) → None
```

Bases: `cabrita.abc.base.InspectTemplate`

GitInspect class.

```
branch_is_dirty (path: str = None) → bool
```

Check if branch is “dirty”.

Ie.: has non-committed modifications.

Parameters `path` – path for branch

Returns bool

```
get_behind_state (path)
```

Check if service need pull and return status.

Parameters `path` – path to search.

Returns string

get_git_revision (*service*)

Return git revision data from service.

Parameters **service** – service name as defined in docker-compose.yml.

Returns string

get_git_revision_from_path (*path, show_branch: bool = False*) → str

Get last tag and most recent commit hash from path.

Parameters

- **path** – path to search
- **show_branch** – check if add branch name to data

Returns string

inspect (*service: str*) → None

Inspect git data from service.

If service are not running from a docker image, try to find the current branch in service path. From this, try to find the relative diff between this branch and target branch

Parameters **service** – service name as defined in docker-compose.yml.

Returns None

1.12 cabrita.components.watchers

Watchers module.

class cabrita.components.watchers.**DockerComposeWatch** (***kwargs*) → None

Bases: *cabrita.components.watchers.Watch*

Docker Compose Watch class.

Watch for docker-compose file status.

```
class cabrita.components.watchers.SystemWatch (background_color:  
                                                cabrita.components.BoxColor           =  
                                                <BoxColor.black: 16>, compose:  
                                                cabrita.components.config.Compose  
                                                = None, git:  
                                                cabrita.components.git.GitInspect  
                                                = None, docker:  
                                                cabrita.components.docker.DockerInspect  
                                                = None) → None
```

Bases: *cabrita.components.watchers.Watch*

System Watch class.

Watch for system monitors (using psutil).

class cabrita.components.watchers.**UserWatch** (***kwargs*) → None

Bases: *cabrita.components.watchers.Watch*

User Watch class.

Watch for user defined watchers in cabrita.yml.

external

Return watchers using external tools.

Default: []

Returns list

file

Return watchers for file dict.

Default: {}

Returns dict

ping

Return watchers for ping addresses.

Default: {}

Returns dict

```
class cabrita.components.watchers.Watch(background_color: cabrita.components.BoxColor
    = <BoxColor.black: 16>, compose:
    cabrita.components.config.Compose = None,
    git: cabrita.components.git.GitInspect = None,
    docker: cabrita.components.docker.DockerInspect
    = None) → None
```

Bases: *[cabrita.components.box.Box](#)*

Watch class.

Base class for watchers based on Box class.

interval

Return interval in seconds for each update.

Default: 30 seconds.

Returns float

run () → None

Check if watch can update his data and execute the update.

Returns None

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

C

- `cabrita.abc`, 4
- `cabrita.abc.base`, 4
- `cabrita.abc.utils`, 5
- `cabrita.command`, 3
- `cabrita.components`, 6
 - `cabrita.components.box`, 7
 - `cabrita.components.config`, 9
 - `cabrita.components.dashboard`, 11
 - `cabrita.components.docker`, 12
 - `cabrita.components.git`, 13
 - `cabrita.components.watchers`, 14
- `cabrita.versions`, 4

A

add_box() (cabrita.components.dashboard.Dashboard method), 12
 add_path() (cabrita.abc.base.ConfigTemplate method), 4
 add_service() (cabrita.components.box.Box method), 7
 ahead (cabrita.components.git.GitDirection attribute), 13
 all_boxes (cabrita.components.dashboard.Dashboard attribute), 12

B

background_color (cabrita.command.CabritaCommand attribute), 3
 background_color (cabrita.components.box.Box attribute), 7
 background_color (cabrita.components.config.Config attribute), 10
 background_color_value (cabrita.components.config.Config attribute), 10
 base_path (cabrita.abc.base.ConfigTemplate attribute), 5
 behind (cabrita.components.git.GitDirection attribute), 13
 black (cabrita.components.BoxColor attribute), 6
 blue (cabrita.components.BoxColor attribute), 6
 both (cabrita.components.docker.PortDetail attribute), 12
 Box (class in cabrita.components.box), 7
 BoxColor (class in cabrita.components), 6
 boxes (cabrita.components.config.Config attribute), 10
 branch_is_dirty() (cabrita.components.git.GitInspect method), 13

C

cabrita.abc (module), 4
 cabrita.abc.base (module), 4
 cabrita.abc.utils (module), 5
 cabrita.command (module), 3
 cabrita.components (module), 6
 cabrita.components.box (module), 7
 cabrita.components.config (module), 9
 cabrita.components.dashboard (module), 11
 cabrita.components.docker (module), 12

cabrita.components.git (module), 13
 cabrita.components.watchers (module), 14
 cabrita.versions (module), 4
 CabritaCommand (class in cabrita.command), 3
 can_update (cabrita.abc.base.InspectTemplate attribute), 5
 can_update (cabrita.components.box.Box attribute), 7
 categories (cabrita.components.box.Box attribute), 7
 check_version() (in module cabrita.versions), 4
 column (cabrita.components.docker.PortView attribute), 13
 Compose (class in cabrita.components.config), 9
 compose_files (cabrita.components.config.Config attribute), 10
 Config (class in cabrita.components.config), 10
 ConfigTemplate (class in cabrita.abc.base), 4
 cyan (cabrita.components.BoxColor attribute), 6

D

Dashboard (class in cabrita.components.dashboard), 11
 DockerComposeWatch (class in cabrita.components.watchers), 14
 DockerInspect (class in cabrita.components.docker), 12

E

execute() (cabrita.command.CabritaCommand method), 3
 external (cabrita.components.docker.PortDetail attribute), 12
 external (cabrita.components.watchers.UserWatch attribute), 14

F

file (cabrita.components.watchers.UserWatch attribute), 15
 format_color() (in module cabrita.abc.utils), 5
 format_revision() (cabrita.components.box.Box static method), 7

G

`generate_boxes()` (cabrita.components.config.Config method), 10
`get_behind_state()` (cabrita.components.git.GitInspect method), 13
`get_build_path()` (cabrita.components.config.Compose method), 9
`get_compose_path()` (cabrita.components.config.Config static method), 11
`get_from_service()` (cabrita.components.config.Compose method), 9
`get_git_revision()` (cabrita.components.git.GitInspect method), 13
`get_git_revision_from_path()` (cabrita.components.git.GitInspect method), 14
`get_path()` (in module cabrita.abc.utils), 5
`get_ports_from_service()` (cabrita.components.config.Compose method), 9
`get_sentry_client()` (in module cabrita.abc.utils), 5
`GitDirection` (class in cabrita.components.git), 13
`GitInspect` (class in cabrita.components.git), 13
`grey` (cabrita.components.BoxColor attribute), 6

H

`has_a_valid_compose` (cabrita.command.CabritaCommand attribute), 3
`has_a_valid_config` (cabrita.command.CabritaCommand attribute), 3
`hidden` (cabrita.components.docker.PortView attribute), 13

I

`ignore_services` (cabrita.components.config.Config attribute), 11
`includes` (cabrita.components.box.Box attribute), 7
`inspect()` (cabrita.abc.base.InspectTemplate method), 5
`inspect()` (cabrita.components.docker.DockerInspect method), 12
`inspect()` (cabrita.components.git.GitInspect method), 14
`InspectTemplate` (class in cabrita.abc.base), 5
`internal` (cabrita.components.docker.PortDetail attribute), 12
`interval` (cabrita.components.box.Box attribute), 8
`interval` (cabrita.components.watchers.Watch attribute), 15
`is_image()` (cabrita.components.config.Compose method), 10
`is_valid` (cabrita.abc.base.ConfigTemplate attribute), 5
`is_valid` (cabrita.components.config.Compose attribute), 10
`is_valid` (cabrita.components.config.Config attribute), 11

L

`layout` (cabrita.components.config.Config attribute), 11
`load_data()` (cabrita.components.box.Box method), 8
`load_file_data()` (cabrita.abc.base.ConfigTemplate method), 5

M

`main` (cabrita.components.box.Box attribute), 8

N

`networks` (cabrita.components.config.Compose attribute), 10

P

`ping` (cabrita.components.watchers.UserWatch attribute), 15
`port_detail` (cabrita.components.box.Box attribute), 8
`port_view` (cabrita.components.box.Box attribute), 8
`PortDetail` (class in cabrita.components.docker), 12
`PortView` (class in cabrita.components.docker), 12
`prepare_dashboard()` (cabrita.command.CabritaCommand method), 4

R

`read_compose_files()` (cabrita.command.CabritaCommand method), 4
`run()` (cabrita.components.box.Box method), 8
`run()` (cabrita.components.dashboard.Dashboard method), 12
`run()` (cabrita.components.watchers.Watch method), 15
`run_command()` (in module cabrita.abc.utils), 5

S

`services` (cabrita.components.box.Box attribute), 8
`services` (cabrita.components.config.Compose attribute), 10
`show_git` (cabrita.components.box.Box attribute), 8
`show_revision` (cabrita.components.box.Box attribute), 8
`size` (cabrita.components.box.Box attribute), 8
`status` (cabrita.components.docker.PortView attribute), 13
`status()` (cabrita.abc.base.InspectTemplate method), 5
`SystemWatch` (class in cabrita.components.watchers), 14

T

`title` (cabrita.components.box.Box attribute), 9
`title` (cabrita.components.config.Config attribute), 11

U

`update_box()` (in module cabrita.components.box), 9
`UserWatch` (class in cabrita.components.watchers), 14

V

`version` (cabrita.abc.base.ConfigTemplate attribute), 5

`versions()` (in module `cabrita.versions`), [4](#)
`volumes` (`cabrita.components.config.Compose` attribute),
[10](#)

W

`Watch` (class in `cabrita.components.watchers`), [15](#)
`watchers` (`cabrita.components.config.Config` attribute), [11](#)
`white` (`cabrita.components.BoxColor` attribute), [6](#)
`widget` (`cabrita.components.box.Box` attribute), [9](#)

Y

`yellow` (`cabrita.components.BoxColor` attribute), [6](#)