
command-executor

Mar 20, 2021

Contents

1	Usage	3
2	Requirements	5
3	Prepare for development	7
4	All Contents	9
4.1	Usage	9
4.2	API Reference	9
4.3	Changelog	10
5	Indices and tables	11
	Python Module Index	13
	Index	15

CHAPTER 1

Usage

Please refer to the [Documentation](#) to learn how to use `command-executor`. Basicly, `command_executor` provides some Python classes to make it easier to start processes / commands and handling errors and output. In addition, input validation for command parameters is possible.

CHAPTER 2

Requirements

python-command-executor supports Python 3 only. No other dependencies are required.

CHAPTER 3

Prepare for development

A Python 3.6 interpreter is required in addition to pipenv.

```
$ pipenv install --python 3.6
$ pipenv shell
$ pip install -e .
```

Now you're ready to run the tests:

```
$ pipenv run py.test
```


4.1 Usage

Note: TODO: Add usage examples for command classes.

4.2 API Reference

4.2.1 Command

```
class command_executor.command.Command(**kwargs)
    Bases: object

    process = None
    command = 'true'
    ignore_output = True
    fail_silently = False
    required_parameters = None
    stdout = -1
    stderr = -1

    execute(ignore_output=None, fail_silently=None, stdin=None, **kwargs)
    validate_parameters()
    get_parameters()
    get_command()
```

```
handle_output (output)
pid
```

4.2.2 Exceptions

exception `command_executor.exceptions.CommandError`

Bases: `Exception`

exception `command_executor.exceptions.CommandExecutionError` (*code*, *stderr*, *command*)

Bases: `command_executor.exceptions.CommandError`

exception `command_executor.exceptions.CommandParameterError`

Bases: `command_executor.exceptions.CommandError`

4.3 Changelog

4.3.1 0.1.0 - 2019-01-10

- Improved support for background / forking processes.

4.3.2 0.0.1 - 2018-08-08

- Initial release of *command-executor*

CHAPTER 5

Indices and tables

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

C

`command_executor.command`, [9](#)

`command_executor.exceptions`, [10](#)

C

`Command` (class in `command_executor.command`), 9
`command` (`command_executor.command.Command` attribute), 9
`command_executor.command` (module), 9
`command_executor.exceptions` (module), 10
`CommandError`, 10
`CommandExecutionError`, 10
`CommandParameterError`, 10

E

`execute()` (`command_executor.command.Command` method), 9

F

`fail_silently` (`command_executor.command.Command` attribute), 9

G

`get_command()` (`command_executor.command.Command` method), 9
`get_parameters()` (`command_executor.command.Command` method), 9

H

`handle_output()` (`command_executor.command.Command` method), 9

I

`ignore_output` (`command_executor.command.Command` attribute), 9

P

`pid` (`command_executor.command.Command` attribute), 10

`process` (`command_executor.command.Command` attribute), 9

R

`required_parameters` (`command_executor.command.Command` attribute), 9

S

`stderr` (`command_executor.command.Command` attribute), 9
`stdout` (`command_executor.command.Command` attribute), 9

V

`validate_parameters()` (`command_executor.command.Command` method), 9