

---

**command-executor**

**unknown**

**Mar 11, 2023**



# CONTENTS

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







## 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.





## REQUIREMENTS

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



## PREPARE FOR DEVELOPMENT

A Python 3 interpreter is required in addition to poetry.

```
$ poetry install
```

Now you're ready to run the tests:

```
$ make tests
```



## ALL CONTENTS

### 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)
    property pid
```

## 4.2.2 Exceptions

**exception** `command_executor.exceptions.CommandError`

Bases: `Exception`

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

Bases: `CommandError`

**exception** `command_executor.exceptions.CommandParameterError`

Bases: `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*

## INDICES AND TABLES

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





## PYTHON MODULE INDEX

### C

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

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



## INDEX

### 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

### M

module  
    `command_executor.command`, 9  
    `command_executor.exceptions`, 10

### P

`pid` (`command_executor.command.Command` property), 9  
`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