
yalow

Release 0.01

Zechariah Thurman

Mar 14, 2020

YALOW

1	Yet Another LOGging Wrapper	1
1.1	Overview	1
1.2	Installation	1
1.3	How Does Do	1
2	Indices and tables	5
	Python Module Index	7
	Index	9

YET ANOTHER LOGGING WRAPPER

1.1 Overview

This package provides a generic wrapper for Python's `logging` module. A log directory is created in the project root directory. The log generated by yalow is intended for use in situations where you need a single project log that all of the packages in your project write to.

1.2 Installation

```
pip install yalow
```

1.3 How Does Do

Get the root path of your project in whatever way you would like. Once that's accomplished provide it to Yalow along with your project name:

```
from yalow import Yalow
logger = Yalow(root_path=I_AM_GROOT, project_name='logalog').logger
```

Example log output format:

```
2020-03-06 21:16:13,495 - logalog - INFO: Logging initialized for project: logalog
2020-03-06 21:16:13,495 - logalog - INFO: IT'S ALIVE!!!!
2020-03-06 21:16:13,496 - logalog.example_package - ERROR: And its minion is HUGE!
```

Refer to examples for usage details.

1.3.1 Examples

The two files below illustrate a basic use case for yalow. They can be found in the examples directory of the yalow github repository for further reference.

logalog.py

```
from pathlib import Path
from yalow import Yalow
from examples.example_package import super_mega_ukulele

if __name__ == '__main__':
    I_AM_GROOT = Path(__file__).parent.parent

    logger = Yalow(root_path=I_AM_GROOT, project_name='logalog').logger

    logger.info(f'IT\'S ALIVE!!!!!!')
    super_mega_ukulele()
```

examples.example_package

```
import logging

module_logger = logging.getLogger('logalog.example_package')

def super_mega_ukulele():
    module_logger.error(f'And its minion is HUGE!')
```

Here's the output that's generated by the code above:

root_path/log/logalog.log

```
2020-03-11 16:43:38,507 - logalog - INFO: Logging initialized for project: logalog
2020-03-11 16:43:38,508 - logalog - INFO: IT'S ALIVE!!!!!!
2020-03-11 16:43:38,508 - logalog.example_package - ERROR: And its minion is HUGE!
```

1.3.2 yalow API

```
class yalow.Yalow(root_path, project_name='yalow', log_dir_name='log', level=10,
                  file_write_mode='w')
    Yet Another LOGging Wrapper
```

Wraps Python's logging module functionality for the generic use case of generating and writing to a shared project log file in the 'root_path/log' directory.

Parameters

- **root_path** (*pathlib.Path*) – Root path of project.
- **project_name** (*str*) – Name of project (The default is 'yalow', I know, very imaginative).

- **log_dir_name** (*str*) – Name of the log directory created in project root (The default is 'log', because as we've already established my imagination is a veritable cornucopia of originality).
- **level** (*logging.Logger.level*) – Default logging level for events added to log (The default is logging.DEBUG).
- **file_write_mode** (*str*) – Mode for writing log file, 'w' for overwrite, 'a' for append (The default behavior is 'w' for overwrite).

log_filepath

Path for generated log file for project.

Type pathlib.Path

logger

Project shared logger object.

Type logging.Logger

handler

Responsible for writing the project log to file.

Type logging.FileHandler

formatter

Responsible for formatting project log file output.

Type logging.Formatter

INDICES AND TABLES

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

PYTHON MODULE INDEX

y

yalow, [2](#)

INDEX

F

`formatter` (*yalow.Yalow attribute*), 3

H

`handler` (*yalow.Yalow attribute*), 3

L

`log_filepath` (*yalow.Yalow attribute*), 3

`logger` (*yalow.Yalow attribute*), 3

Y

`Yalow` (*class in yalow*), 2

`yalow` (*module*), 2