
aelog Documentation

Release stable

Sep 27, 2020

Contents

1	Installing aelog	3
2	init aelog	5
3	aelog config	7
4	Usage	9
4.1	simple using, output log to terminal.	9
4.2	output log to file and terminal.	10
4.3	asynchronous output log to file and terminal.	10

An simple, async, full package name path, log rotating, different colored log library.

aelog aims to make using python log as simple as possible. as a result, it drastically simplifies using python logging.

aelog's design objectives:

- Make using python log as simple as possible.
- Output log contains the full package name path.
- Provide asynchronous log output function, at the same time, contains common log output.
- Output according to the log level to mark the different colors separately.
- Provide a log file rotating, automatic backup.
- Output to the terminal and file, default output to the terminal, if you don't provide the log file path.

CHAPTER 1

Installing aelog

- `pip install aelog`

CHAPTER 2

init aelog

```
import aelog

app = Flask(__name__)

aelog.init_app(app)
# or
aelog.init_app(aelog_access_file='aelog_access_file.log', aelog_error_file='aelog_
↪error_file.log',
               aelog_console=False)
```


List of configuration keys that the aelog extension recognizes:

configuration key	the meaning of the configuration key
AELOG_ACCESS_FILE	Access file path, default None.
AELOG_ERROR_FILE	Error file path, default None.
AELOG_CONSOLE	Whether it is output at the terminal, default false.
AELOG_MAX_BYTES	Log file size, default 50M.
AELOG_BACKUP_COUNT	Rotating file count, default 5.

4.1 simple using, output log to terminal.

```
import aelog

aelog.init_app(aelog_console=True)

def test_aelog_output_console():
    """
    Args:

    Returns:

    """
    aelog.debug("simple debug message", "other message")
    aelog.info("simple info message", "other message")
    aelog.warning("simple warning message", "other message")
    aelog.error("simple error message", "other message")
    aelog.critical("simple critical message", "other message")
    try:
        5 / 0
    except Exception as e:
        aelog.exception(e)
```

This will output to the terminal.

```
[2020-06-15 14:29:09 +0800] [DEBUG] __main__ [test_aelog_output_console 43]: simple debug message other message
[2020-06-15 14:29:09 +0800] [INFO] __main__ [test_aelog_output_console 44]: simple info message other message
[2020-06-15 14:29:09 +0800] [WARNING] __main__ [test_aelog_output_console 45]: simple warning message other message
[2020-06-15 14:29:09 +0800] [ERROR] __main__ [test_aelog_output_console 46]: simple error message other message
[2020-06-15 14:29:09 +0800] [CRITICAL] __main__ [test_aelog_output_console 47]: simple critical message other message
[2020-06-15 14:29:09 +0800] [ERROR] __main__ [test_aelog_output_console 51]: division by zero
Traceback (most recent call last):
  File "/home/guoyanfeng/github/aelog/tests/test_aelog.py", line 49, in test_aelog_output_console
    5 / 0
ZeroDivisionError: division by zero
```

- Different levels of logging, different color, the color is cyan, green, yellow, red and 'bold_red,bg_white' in turn.

4.2 output log to file and terminal.

```
import aelog
from flask import Flask

app = Flask(__name__)

aelog.init_app(app) # Output to the test.log file and terminal

def test_aelog_output_file():
    """
    Args:

    Returns:

    """
    aelog.debug("simple debug message", "other message")
    aelog.info("simple info message", "other message")
    aelog.warning("simple warning message", "other message")
    aelog.error("simple error message", "other message")
    aelog.critical("simple critical message", "other message")
    try:
        5 / 0
    except Exception as e:
        aelog.exception(e)
```

```
[2020-06-15 14:32:50 +0800] [DEBUG] __main__ [test_aelog_output_file 63]: simple debug
[2020-06-15 14:32:50 +0800] [INFO] __main__ [test_aelog_output_file 64]: simple info m
[2020-06-15 14:32:50 +0800] [WARNING] __main__ [test_aelog_output_file 65]: simple wa
[2020-06-15 14:32:50 +0800] [ERROR] __main__ [test_aelog_output file 66]: simple error
[2020-06-15 14:32:50 +0800] [CRITICAL] __main__ [test_aelog_output_file 67]: simple cr
[2020-06-15 14:32:50 +0800] [ERROR] __main__ [test_aelog_output_file 71]: division by
Traceback (most recent call last):
  File "/home/guoyanfeng/github/aelog/tests/test_aelog.py", line 69, in test_aelog_outp
    5 / 0
ZeroDivisionError: division by zero
```

This will output to the test.log file and terminal.

- Automatic output is greater than the error information to the 'test_error.log' file. - Different levels of logging, different color, the color is cyan, green, yellow, red and 'bold_red,bg_white' in turn.

4.3 asynchronous output log to file and terminal.

```
import asyncio
import aelog
```

(continues on next page)

(continued from previous page)

```

from sanic import Sanic

app = Sanic(__name__)

aelog.init_aelog(app) # Output to the test.log file and terminal

async def test_async_output():
    await aelog.async_debug("simple debug message", "other message")
    await aelog.async_info("simple info message", "other message")
    await aelog.async_warning("simple warning message", "other message")
    await aelog.async_error("simple error message", "other message")
    await aelog.async_critical("simple critical message", "other message")
    try:
        5 / 0
    except Exception as e:
        await aelog.async_exception(e)

if "__name__"=="__main__":
    loop = asyncio.get_event_loop()
    loop.run_until_complete(test_async_output())

```

This will output to the test.log file and terminal.

```

[2020-06-15 14:33:39 +0800] [DEBUG] __main__ [test_async_output 77]: simple debug message other message
[2020-06-15 14:33:39 +0800] [INFO] __main__ [test_async_output 78]: simple info message other message
[2020-06-15 14:33:39 +0800] [WARNING] __main__ [test_async_output 79]: simple warning message other message
[2020-06-15 14:33:39 +0800] [ERROR] __main__ [test_async_output 80]: simple error message other message
[2020-06-15 14:33:39 +0800] [CRITICAL] __main__ [test_async_output 81]: simple critical message other message
[2020-06-15 14:33:39 +0800] [ERROR] __main__ [test_async_output 85]: division by zero
Traceback (most recent call last):
  File "/home/guoyanfeng/github/aelog/tests/test_aelog.py", line 83, in test_async_output
    5 / 0
ZeroDivisionError: division by zero

```

- Automatic output is greater than the error information to the 'test_error.log' file.
- Different levels of logging, different color, the color is cyan, green, yellow, red and 'bold_red,bg_white' in turn.