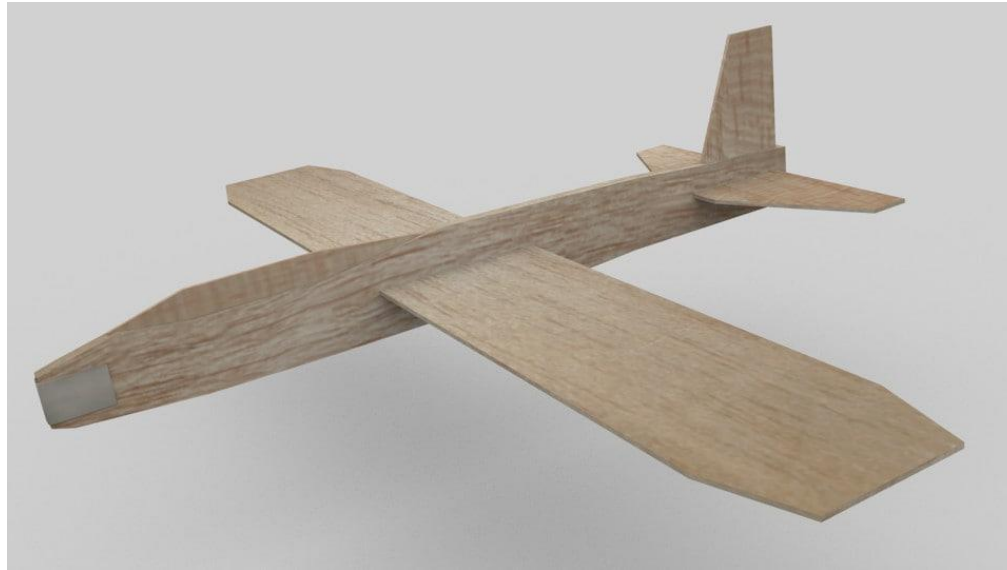


# Balsa

## Lightweight Python Logging



<https://www.turbosquid.com/3d-models/blend-balsa-wood-toy-airplane/608014>

James Abel

PyBay - Sept 10, 2022

[j@abel.co](mailto:j@abel.co)

# The logging module

- The logging module is awesome!
  - Handlers – stream (console), files, sockets, HTTP, custom, ... many more!
  - Filters
  - Formatters
  - Hierarchal
  - Log Levels – debug, info, warning, error, critical
- `logging.getLogger(name)` provides the logger associated with `name`
  - Can directly access a logger from anywhere in your program with just the `name` string
- <https://docs.python.org/library/logging.html>

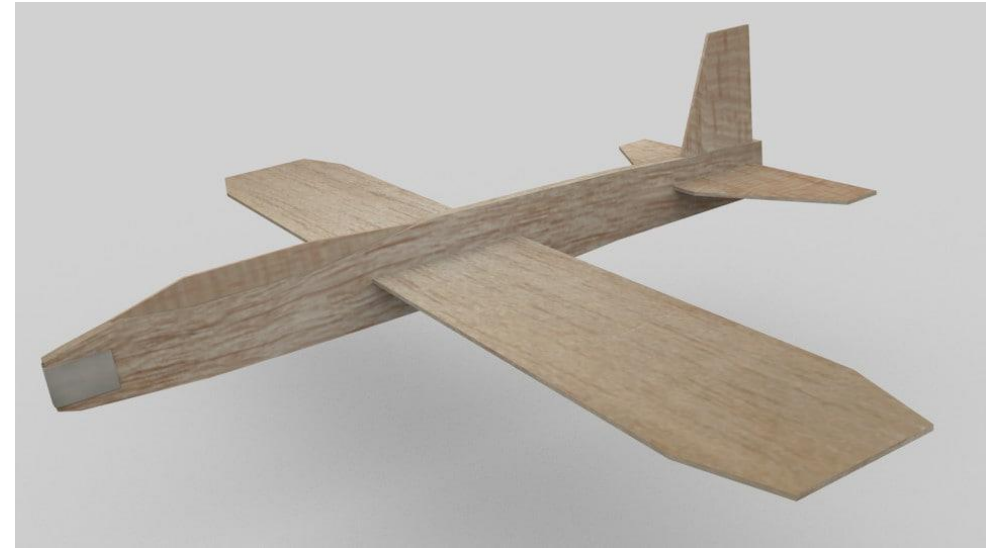
# However, logging options and configurations can get rather involved for relatively simple apps

- Log message format
- Handlers
  - UI, File, Services
- Where to write log files?
- Log levels
  - Often different for each handler
- GUI
- Multiprocessing
- Tracebacks
- Interface into cloud services

**Setting up logging Can Be Significant Effort**

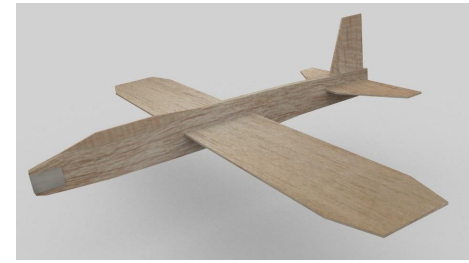
# Balsa – Lightweight Logging

- Provide useful logging with a few lines of code
- Consistent formatting and interface
  - `appdirs` for log file directory
- CLI and GUI
  - GUI: `tkinter` or `PyQt5` dialog box (avoid writing to `stdout/stderr`)
- Structured Logging (via `yasf`)
- Supports Multiprocessing
  - Balsa → config as a dict → Balsa clone in separate process
- Object Oriented
  - Inherit from `Balsa` to select options, set alternative log levels, provide service credentials, etc.
- Optional Services
  - AWS CloudWatch (via `boto3`) – query-able structured logging
  - Sentry exception service - e.g., for `>= ERROR` log level (`sentry-sdk`)
- Verbosity expressed by intent rather than level (e.g., `verbose` for development)
- Cross-platform
- Available on PyPI



**`pip install balsa`**

# Simple Example



```
from balsa import get_logger, Balsa
```

```
application_name = 'example'  
log = get_logger(application_name)
```

```
def main():
```

```
    balsa = Balsa(name=application_name, author='james abel')  
    balsa.init_logger()
```

```
    log.error('my error example')
```

Output:

2018-08-18 20:43:33,756 - example - balsa\_simple\_example.py - 12 - main - ERROR - my error example

Annotations:

- timestamp (points to 2018-08-18 20:43:33,756)
- app name (points to example)
- source file name (points to balsa\_simple\_example.py)
- line number (points to 12)
- level (points to ERROR)
- message (points to my error example)

Also writes out a file (e.g., Windows):

C:\Users\<user>\AppData\Local\james abel\example\Logs\example.log

**CloudWatch**

**Logs Insights**

Select log groups, and then run a query or choose a sample query.

5m 30m **1h** 3h 12h Custom

Select log group(s)

test\_balsa\_aws\_cloudwatch\_logs X

```

1 fields @timestamp, @message
2 | filter levelname="WARNING"
3 | sort @timestamp desc
4 | limit 20

```

Run query Save Actions History

Queries are allowed to run for up to 15 minutes.

Logs Visualization Export results Add to dashboard

Showing 3 of 3 records matched

3 records (1.8 kB) scanned in 2.5s @ 1 records/s (716.222 B/s)

#	@timestamp	@message
1	2022-09-08T11:53:57.19...	{"is_mock": false, "issue": "something really went wrong", "message": "another mes

Field Value

@ingestionTime 1662663237620

@log 076966278319:test\_balsa\_aws\_cloudwatch\_logs

@logStream jamespersonalvm1-james

@message {"is\_mock": false, "issue": "something really went wrong", "message": "another message", "creat

@timestamp 1662663237192

created 1.6626632367193263E9

filename test\_balsa\_aws\_cloudwatch\_logs.py

funcName test\_balsa\_aws\_cloudwatch\_logs

is\_mock 0

issue something really went wrong

levelname WARNING

lineno 21

message another message

module test\_balsa\_aws\_cloudwatch\_logs

name test\_balsa\_aws\_cloudwatch\_logs

pathname C:\Users\james\projects\balsa\test\_balsa\_aws\_cloudwatch\_logs.py

process 11772

processName MainProcess

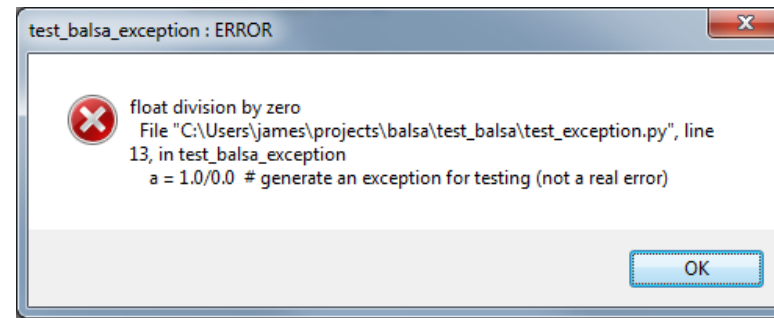
system\_computer\_name jamespersonalvm1

system\_user\_name james

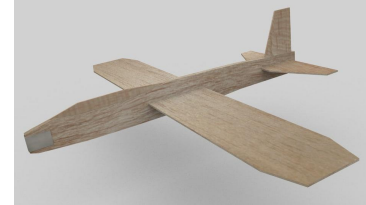
thread 7680

threadName MainThread

## AWS CloudWatch



## GUI messages (rate limited)



[View on Sentry](#)

**SENTRY**

**New alert from balsa**

ISSUE

**error** test balsa sentry error message

ID:

Aug. 19, 2018, 6:25:42 a.m. UTC

**Message**

test balsa sentry error message

**Tags**

level = error

logger = test\_balsa\_sentry

server\_name =

You are receiving this email due to matching rules: [Send a notification for new issues](#)

## Sentry (exception service)

# Summary and Thank You

- Try balsa!  
`pip install balsa`  
<https://github.com/jamesabel/balsa>  
<http://balsa.readthedocs.io/>
- Please provide feedback, issues, PRs, ...

