
marquee Documentation

Release unknown

ClearDATA

Oct 21, 2016

1	marquee	3
1.1	Features	3
1.2	Documentation	3
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	Usage	7
3.1	Handler	7
3.2	Formatter	7
4	Contributing	9
4.1	Get Started	9
4.2	Pull / Merge Request Guidelines	9
5	Credits	11
5.1	Core Developers	11
5.2	Contributors	11
6	marquee	13
6.1	marquee package	13
6.1.1	Submodules	13
6.1.2	marquee.formatter module	13
6.1.3	marquee.handler module	13
6.1.4	Module contents	13
7	Indices and tables	15
	Python Module Index	17

Contents:

marquee is a very simply Python logging formatter and handler to send log events as CloudWatch Events.

1.1 Features

- simple log formatter
- handler to send events to CloudWatch Events

1.2 Documentation

Documentation can be found at <http://marquee.readthedocs.io/en/latest/>.

Installation

2.1 Stable release

To install marquee, run this command in your terminal:

```
$ pip install marquee
```

This is the preferred method to install, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for marquee can be downloaded from the [Github repo](#).

Clone the repository:

```
$ git clone git@github.com:theherk/marquee.git
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

Usage

3.1 Handler

To use the handler, simply add it to you log:

```
import logging

from marquee.formatter import MarqueeFormatter
from marquee.handler import CloudWatchEventsHandler

log = logging.getLogger(__name__)
log.setLevel(logging.DEBUG)

handler = CloudWatchEventsHandler()
log.addHandler(handler)
```

3.2 Formatter

To use the formatter, which is optional, add it to your handler. If you don't use this formatter, `__name__` will be used for the source in the handler. If you do, you can pass *source* too the handler along with any other formatter arguments. Then, this will be included in the CloudWatch Event.

To add the formatter to the handler:

```
fmt = MarqueeFormatter('theherk.testapp')
handler.setFormatter(fmt)
```

Contributing

Contributions are welcome. You can contribute in the following ways.

- report bugs
- fix bugs
- implement features
- write documentation
- submit feedback

4.1 Get Started

4.2 Pull / Merge Request Guidelines

Credits

5.1 Core Developers

- Herkermer Sherwood theherk@gmail.com

5.2 Contributors

6.1 marquee package

6.1.1 Submodules

6.1.2 marquee.formatter module

marquee logging formatter.

```
class marquee.formatter.MarqueeFormatter (source=None, *args, **kwargs)
```

Bases: logging.Formatter

```
format (record)
```

Overridden format method.

This will create a json object containing data for the CloudWatch Event.

```
class marquee.formatter.MarqueeEventFormatter (event_type=None, source=None, *args, **kwargs)
```

Bases: logging.Formatter

```
format (record)
```

Overridden format method. This will create a json object containing data for the CloudWatch Event.

6.1.3 marquee.handler module

marquee logging handler.

```
class marquee.handler.CloudWatchEventsHandler (session=None, detail_type=None, *args, **kwargs)
```

Bases: logging.Handler

```
emit (record)
```

Overridden emit method.

This will emit the log message to CloudWatch Events. Permissions and target account all set according to normal boto resolution.

6.1.4 Module contents

Indices and tables

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

m

`marquee`, [13](#)

`marquee.formatter`, [13](#)

`marquee.handler`, [13](#)

C

CloudWatchEventsHandler (class in `marquee.handler`),
[13](#)

E

`emit()` (`marquee.handler.CloudWatchEventsHandler`
method), [13](#)

F

`format()` (`marquee.formatter.MarqueeEventFormatter`
method), [13](#)

`format()` (`marquee.formatter.MarqueeFormatter` method),
[13](#)

M

`marquee` (module), [13](#)

`marquee.formatter` (module), [13](#)

`marquee.handler` (module), [13](#)

`MarqueeEventFormatter` (class in `marquee.formatter`), [13](#)

`MarqueeFormatter` (class in `marquee.formatter`), [13](#)