
cdumay-opentracing Documentation

Release 0.0.1

Cedric Dumay

Feb 19, 2021

Contents

1	Example	3
2	Get It Now	7
3	Features & API Focus	9
	Index	13

cdumay-opentracing is a python library to facilitate opentracing integration.

CHAPTER 1

Example

The following example shows how to use it with Jaeger:

```
1 import opentracing
2 from cdumay_opentracing import OpenTracingDriver, OpenTracingSpan
3
4
5 class A(object):
6     "A sample class"
7     def __init__(self):
8         self.a = 5
9         self.b = "toto"
10        self.trace = dict()
11
12
13 class B(A):
14     """A class which inherit of A"""
15
16
17 class DriverA(OpenTracingDriver):
18     @classmethod
19     def extract(cls, data):
20         return opentracing.tracer.extract(cls.FORMAT, data.trace)
21
22     @classmethod
23     def inject(cls, span, data):
24         opentracing.tracer.inject(span, cls.FORMAT, data.trace)
25
26     @classmethod
27     def tags(cls, data):
28         return dict(a=data.a, b=data.b)
29
30
31 def child():
32     """A function that automatically recovers the current span"""
```

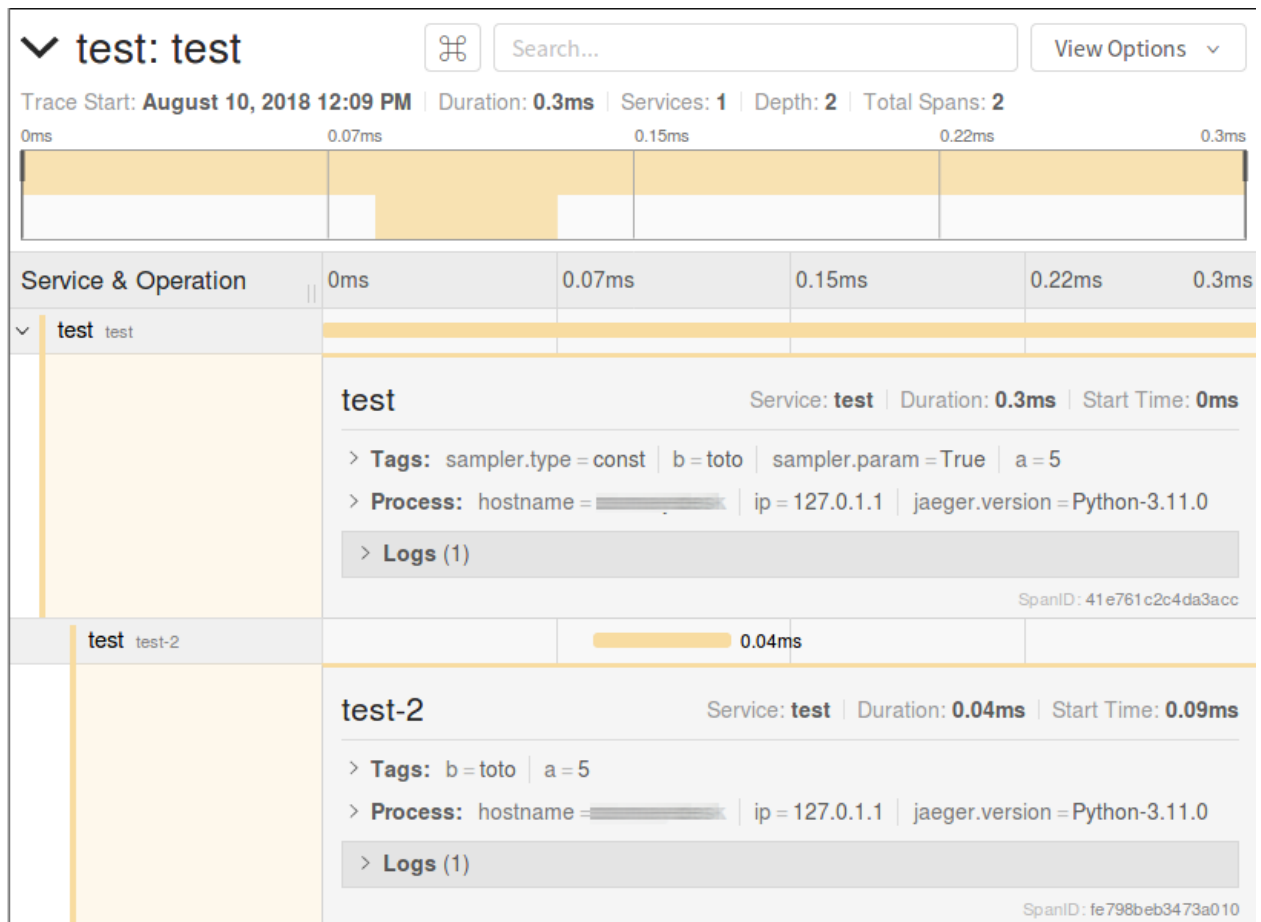
(continues on next page)

(continued from previous page)

```
33     with OpenTracingSpan(B(), "test-2") as span:
34         span.log_kv(dict(event="done"))
35
36
37 if __name__ == '__main__':
38     import os, time
39     from jaeger_client import Config
40     from cdumay_opentracing import OpenTracingManager
41
42     tracer = Config(service_name="test", config=dict(
43         sampler=dict(type='const', param=1), logging=True,
44         local_agent=dict(reporting_host=os.getenv('JAEGER_HOST', 'localhost'))
45     )).initialize_tracer()
46
47     OpenTracingManager.register(A, DriverA)
48
49     a = A()
50     with OpenTracingSpan(a, "test") as span:
51         child()
52         OpenTracingManager.log_kv(span, a, "done")
53
54     time.sleep(4)
55     tracer.close()
```

1.1 Explanations

- **line 5-14:** We create classes for our example.
- **line 17-28:** We create a driver which allows to automate the manipulation of the class A.
- **line 38-45:** We initialize tracing
- **line 47:** We register the driver into the Tracing manager. It will allow to manipulate A instances (and by extension B).
- **line 49-50:** We make a span using a.
- **line 51:** We call a function from a span which will create a sub-span.
- **line 54-55:** We sleep and close to make sure that the span is sent.



CHAPTER 2

Get It Now

Install the lib using `pip`:

```
pip install -U cdumay-opentracing
```


3.1 OpenTracingManager — The trace manager

class `cdumay_opentracing.OpenTracingManager`

This class manages the opentracing context.

- It allows to recover using the `opentracing` library the initialized tracer.
- It manage spans stack

classmethod `register (clazz, driver)`

Register a driver for the given class

Parameters

- **clazz** (*Any*) – Class managed by the driver.
- **driver** (`cdumay_opentracing.OpenTracingDriver`) – Driver to register.

classmethod `get_driver (obj)`

Find registered driver for the given object

Parameters **obj** (*Any*) – object to manipulate

Returns registered driver for this class

Return type `cdumay_opentracing.OpenTracingDriver`

classmethod `tags (obj)`

Extract tags from *carrier* object using a registered driver.

Parameters **obj** (*Any*) – object to manipulate

Returns Tags to add on span

Return type dict

classmethod `get_current_span ()`

Get the current span

Returns The current span

Return type `opentracing.span.Span` or `None`

classmethod `create_span` (*obj*, *name*, *tags*)

Create a new span

Parameters

- **obj** (*Any*) – Any object
- **name** (*str*) – Span name
- **tags** (*dict*) – Additional tags

Returns `Span`

Return type `opentracing.span.Span`

classmethod `finish_span` (*span*)

Terminate the given span

Parameters **span** (*opentracing.span.Span*) – Span to finish

classmethod `log_kv` (*span*, *obj*, *event*, ***kwargs*)

Adds a log record to the Span.

Parameters

- **span** (*opentracing.span.Span*) – the Span instance to use.
- **obj** (*Any*) – the *carrier* object.
- **event** (*str*) – Span event name.
- **kwargs** (*dict*) – A dict of string keys and values of any type to log

Returns Returns the Span itself, for call chaining.

Return type `opentracing.span.Span`

3.2 OpenTracingDriver - Classes driver

class `cdumay_opentracing.OpenTracingDriver`

FORMAT

`opentracing.Format` used to load and store span context

classmethod `extract` (*data*)

Extract span context from a *carrier* object

Parameters **data** (*Any*) – the *carrier* object.

Returns a `SpanContext` instance extracted from *carrier* or `None` if no such span context could be found.

classmethod `inject` (*span*, *data*)

Injects the span context into a *carrier* object.

Parameters

- **span** (*opentracing.span.SpanContext*) – the SpanContext instance to inject
- **data** (*Any*) – the *carrier* object.

classmethod tags (*data*)

Extract tags from *carrier* object.

Parameters **data** (*Any*) – the *carrier* object.

Returns Tags to add on span

Return type dict

classmethod log_kv (*span, data, event, **kwargs*)

Adds a log record to the Span.

Parameters

- **span** (*opentracing.span.Span*) – the Span instance to use.
- **data** (*Any*) – the *carrier* object.
- **event** (*str*) – Span event name.
- **kwargs** (*dict*) – A dict of string keys and values of any type to log

Returns Returns the Span itself, for call chaining.

Return type opentracing.span.Span

C

`cdumay_opentracing.OpenTracingDriver`
(built-in class), 10

`cdumay_opentracing.OpenTracingManager`
(built-in class), 9

`create_span()` (cdumay_opentracing.OpenTracingManager
class method), 10

E

`extract()` (cdumay_opentracing.OpenTracingDriver
class method), 10

F

`finish_span()` (cdumay_opentracing.OpenTracingManager
class method), 10

`FORMAT` (cdumay_opentracing.OpenTracingDriver attribute), 10

G

`get_current_span()` (cdumay_opentracing.OpenTracingManager
class method), 9

`get_driver()` (cdumay_opentracing.OpenTracingManager
class method), 9

I

`inject()` (cdumay_opentracing.OpenTracingDriver
class method), 10

L

`log_kv()` (cdumay_opentracing.OpenTracingDriver
class method), 11

`log_kv()` (cdumay_opentracing.OpenTracingManager
class method), 10

R

`register()` (cdumay_opentracing.OpenTracingManager
class method), 9

T

`tags()` (cdumay_opentracing.OpenTracingDriver class
method), 11

`tags()` (cdumay_opentracing.OpenTracingManager
class method), 9