base10

Release 0.5.3

Jul 31, 2017

Contents

Base10 is a metrics abstractoin layer for linking multiple metrics source and stores. It also simplifies metric creation and proxying.

CHAPTER 1

Installing Base10

Base10 can be installed from Pypi using pip:

```
pip install base10
```

CHAPTER 2

Using Base10

Example

This shows a simple metric generator that writes a JSON formatted metric, containing a random value, to RabbitMQ.

```
from random import random
from time import sleep

from base10 import MetricHelper, MetricHandler
from base10.dialects import JSONDialect
from base10.transports import RabbitMQWriter

if __name__ == '__main__':

    class MyMetric(MetricHelper):
        _name = 'metric'

        _fields = [
            'value',
        ]

        _metadata = [
            'hostname',
        ]

    class JSON(MetricHandler):
        _dialect = JSONDialect()
        _writer = RabbitMQWriter(
            broker='127.0.0.1', exchange='amq.topic', topic='metrics.example')

    json = JSON()

    while True:
        json.write(MyMetric(value=random(), hostname='test'))
        sleep(1)
```

This shows a simple proxy that reads JSON formatted metrics from RabbitMQ and outputs them in InfluxDB format over a UDP socket.

```
from base10 import MetricHandler
from base10.dialects import JSONDialect, SplunkDialect  #InfluxDBDialect
from base10.transports import RabbitMQReader, UDPWriter

if __name__ == '__main__':

    class RabbitMQ(MetricHandler):
        _dialect = JSONDialect()
        _reader = RabbitMQReader(
            broker='127.0.0.1', exchange='amq.topic', routing_key='metrics.#')

    class InfluxDB(MetricHandler):
        _dialect = SplunkDialect()  #InfluxDBDialect()
        _writer = UDPWriter(host='127.0.0.1', port=10000)

    rabbitmq = RabbitMQ()
    influxdb = InfluxDB()

    for metric in rabbitmq.read():
        influxdb.write(metric)
```