

---

# **base10**

***Release 0.6.1***

**Feb 11, 2018**



---

## Contents

---

<b>1</b>	<b>Guide</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Usage . . . . .	3
1.3	API Reference . . . . .	5
<b>2</b>	<b>Project Info</b>	<b>7</b>
2.1	Changelog . . . . .	7
2.2	License . . . . .	8
	<b>Python Module Index</b>	<b>9</b>



Release v0.6.1. (*[Changelog](#)*)

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



## 1.1 Installation

### 1.1.1 From PyPi

```
$ pip install base10
```

### 1.1.2 From source

**Base10** is developed on [Github](#).

You can clone the public repo:

```
$ git clone https://github.com/mattdavis90/base10.git
```

Once you have the source, you can install it into your site-packages with

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

## 1.2 Usage

Use `MetricHelper` to aid *Metric* creation, and `MetricHandler` to aid reading and writing metrics.

### 1.2.1 Metric Generator

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)
```

## 1.2.2 Metric Proxy

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)
```



## 1.3 API Reference

### 1.3.1 Base Classes

```
class base10.base.Dialect (*args, **kwargs)
```

```
    __init__ (*args, **kwargs)
```

```
    from_string (string)
```

```
    to_string (metric)
```

```
class base10.base.Metric (name, fields, metadata, **kwargs)
```

Generic Metric class

```
    __init__ (name, fields, metadata, **kwargs)
```

Create a new Metric

#### Parameters

- **name** – Name of the metric.
- **fields** – List of field names to include.
- **metadata** – List of metadata field names to include.
- **\*\*kwargs** – Keyword values for the fields and metadata.

#### fields

Get Metric fields

#### metadata

Get Metric metadata

#### name

Get Metric name

#### values

Get Metric values

```
class base10.base.Reader (*args, **kwargs)
```

```
    __init__ (*args, **kwargs)
```

```
    read ()
```

```
class base10.base.Writer (*args, **kwargs)
```

```
    __init__ (*args, **kwargs)
```

```
    write (string)
```

### 1.3.2 Helper CLasses

### 1.3.3 Exceptions

### 1.3.4 Dialects

```
class base10.dialects.json_dialect.JSONDialect (*args, **kwargs)
```

```
{ name: 'cpu_usage', fields: {  
  user: 0.2, free: 0.75  
}, metadata: {  
  hostname: 'host-1'  
}, timestamp: 1489478831  
}
```

### **1.3.5 Transports**

## 2.1 Changelog

### 2.1.1 0.5.3

- **Removed:** Python2.6 support (It didn't work anyway)

### 2.1.2 0.5.2

- **Added:** `base10.__version__`

### 2.1.3 0.5.1 - README on PyPi

- **Changed:** README on PyPi (Again!)

### 2.1.4 0.5.0 - Testing, Docs, Coverage

- **Changed:** Semantic versioning
- **Changed:** Testing using Travis-CI
- **Changed:** Docs on Read the Docs
- **Changed:** Coverage on Coveralls

### 2.1.5 0.5 - Splunk Support

- **Added:** Splunk Dialect (Write-only)
- **Changed:** RabbitMQ kwarg pass-through

- **Changed:** Faster time() function

### 2.1.6 0.4 - README on PyPi

- **Changed:** Added README to PyPi

### 2.1.7 0.3 - RabbitMQ Support

- **Added:** RabbitMQ Transport
- **Changed:** Nanosecond timestamp
- **Changed:** Breaking changes to API

### 2.1.8 0.2 - Influx and UDP Support

- **Added:** InfluxDB Dialect (Write-only)
- **Added:** UDP Transport (Send-only)
- **Changed:** Millisecond timestamp
- **Changed:** Breaking changes to API

### 2.1.9 0.1 - Initial Release

- **Added:** File Transport
- **Added:** JSON Dialect

## 2.2 License

The MIT License (MIT)

Copyright (c) 2017 Matt Davis

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

The above copyright notice **and** this permission notice shall be included **in all** copies **or** substantial portions of the Software.

THE SOFTWARE IS PROVIDED "**AS IS**", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- - `base10.base`, 5
  - `base10.dialects.influxdb_dialect`, 5
  - `base10.dialects.json_dialect`, 5
  - `base10.dialects.splunk_dialect`, 6
  - `base10.exceptions`, 5
  - `base10.helpers`, 5
  - `base10.transports.file_transport`, 6
  - `base10.transports.rabbitmq_transport`, 6
  - `base10.transports.udp_transport`, 6



## Symbols

`__init__()` (base10.base.Dialect method), 5  
`__init__()` (base10.base.Metric method), 5  
`__init__()` (base10.base.Reader method), 5  
`__init__()` (base10.base.Writer method), 5

## B

base10.base (module), 5  
base10.dialects.influxdb\_dialect (module), 5  
base10.dialects.json\_dialect (module), 5  
base10.dialects.splunk\_dialect (module), 6  
base10.exceptions (module), 5  
base10.helpers (module), 5  
base10.transports.file\_transport (module), 6  
base10.transports.rabbitmq\_transport (module), 6  
base10.transports.udp\_transport (module), 6

## D

Dialect (class in base10.base), 5

## F

fields (base10.base.Metric attribute), 5  
from\_string() (base10.base.Dialect method), 5

## J

JSONDialect (class in base10.dialects.json\_dialect), 5

## M

metadata (base10.base.Metric attribute), 5  
Metric (class in base10.base), 5

## N

name (base10.base.Metric attribute), 5

## R

read() (base10.base.Reader method), 5  
Reader (class in base10.base), 5

## T

to\_string() (base10.base.Dialect method), 5

## V

values (base10.base.Metric attribute), 5

## W

write() (base10.base.Writer method), 5  
Writer (class in base10.base), 5