
sprockets-influxdb

Release 2.2.0

Nov 27, 2018

Contents

1	Installation	3
2	Documentation	5
3	Configuration	7
4	Example	9
5	Requirements	11
6	Version History	13
7	License	15

Buffering InfluxDB client and mixin for Tornado applications

CHAPTER 1

Installation

sprockets-influxdb is available on the Python package index and is installable via pip:

```
pip install sprockets-influxdb
```


CHAPTER 2

Documentation

Documentation is available at sprockets-influxdb.readthedocs.io.

CHAPTER 3

Configuration

Configuration can be managed by specifying arguments when invoking `sprockets_influxdb.install` or by using environment variables.

For programmatic configuration, see the [sprockets_influxdb.install](#) documentation.

The following table details the environment variable configuration options.

Variable	Definition	De- fault
INFLUXDB_SCHEME	The URL request scheme for making HTTP requests	https
INFLUXDB_HOST	The InfluxDB server hostname	localhost
INFLUXDB_PORT	The InfluxDB server port	8086
INFLUXDB_USER	The InfluxDB server username	
INFLUXDB_PASSWORD	The InfluxDB server password	
INFLUXDB_ENABLED	Set to false to disable InfluxDB support	true
INFLUXDB_INTERVAL	How many milliseconds to wait before submitting measurements when the buffer has fewer than INFLUXDB_TRIGGER_SIZE measurements.	60000
INFLUXDB_MAX_BATCH	Max # of measurements to submit in a batch	10000
INFLUXDB_MAX_BUFFER	Limit of measurements in a buffer before new measurements are discarded.	25000
INFLUXDB_SAMPLE_PROBABILITY	A value that is ≥ 0 and ≤ 1.0 that specifies the probability that a batch will be submitted to InfluxDB or dropped.	1.0
INFLUXDB_TRIGGER_SIZE	The number of metrics in the buffer to trigger the submission of a batch.	60000
INFLUXDB_TAG_HOSTNAME	Include the hostname as a tag in the measurement	true

3.1 Mixin Configuration

The `sprockets_influxdb.InfluxDBMixin` class will automatically tag the measurement if the `ENVIRONMENT` environment variable is set with the environment that the application is running in. Finally, if you are using the [Sprockets Correlation Mixin](#), measurements will automatically be tagged with the correlation ID for a request.

CHAPTER 4

Example

In the following example, a measurement is added to the example InfluxDB database with the measurement name of `measurement-name`. When the `~tornado.ioloop.IOLoop` is started, the `stop` method is invoked, calling `~sprockets_influxdb.shutdown`. `~sprockets_influxdb.shutdown` ensures that all of the buffered metrics are written before the `IOLoop` is stopped.

```
import logging

import sprockets_influxdb as influxdb
from tornado import ioloop

logging.basicConfig(level=logging.INFO)

io_loop = ioloop.IOLoop.current()
influxdb.install(io_loop=io_loop)

measurement = influxdb.Measurement('example', 'measurement-name')
measurement.set_tag('foo', 'bar')
measurement.set_field('baz', 1.05)

influxdb.add_measurement(measurement)

def stop():
    influxdb.shutdown()
    io_loop.stop()

io_loop.add_callback(stop)
io_loop.start()
```


CHAPTER 5

Requirements

- Tornado

CHAPTER 6

Version History

Available at <https://sprockets-influxdb.readthedocs.org/en/latest/history.html>

Copyright (c) 2016 AWeber Communications All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Sprockets nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7.1 Sprockets InfluxDB API

To use the InfluxDB client, you need to install it into the runtime environment. To do so, you use the `install()` method. Then metrics can be added by creating instances of the `Measurement` class and then adding them to the write buffer by calling `add_measurement()`. In the following example, a measurement is added to the example InfluxDB database with the measurement name of `measurement-name`. When the IOLoop is started, the `stop` method is invoked which calls `shutdown()`. `shutdown()` ensures that all of the buffered metrics are written before the IOLoop is stopped.

Measurements will be sent in batches to InfluxDB when there are `INFLUXDB_TRIGGER_SIZE` measurements in the buffer or after `INFLUXDB_INTERVAL` milliseconds have passed since the last measurement was added, whichever occurs first.

The timeout timer for submitting a buffer of `< INFLUXDB_TRIGGER_SIZE` measurements is only started when there isn't an active timer, there is not a batch currently being written, and a measurement is added to the buffer.

```
import logging

import sprockets_influxdb as influxdb
from tornado import ioloop

logging.basicConfig(level=logging.INFO)

io_loop = ioloop.IOLoop.current()
influxdb.install(io_loop=io_loop)

measurement = influxdb.Measurement('example', 'measurement-name')
measurement.set_tag('foo', 'bar')
measurement.set_field('baz', 1.05)

influxdb.add_measurement(measurement)

def stop():
    influxdb.shutdown()
    io_loop.stop()

io_loop.add_callback(stop)
io_loop.start()
```

If you are writing a Tornado web application, you can automatically instrument all of your Request Handlers by adding the *InfluxDBMixin*:

```
import logging
import os

import sprockets_influxdb as influxdb
from tornado import ioloop, web

class RequestHandler(influxdb.InfluxDBMixin,
                     web.RequestHandler):

    def get(self, *args, **kwargs):
        self.write({'hello': 'world'})

if __name__ == '__main__':
    logging.basicConfig(level=logging.INFO)

    os.environ['ENVIRONMENT'] = 'development'
    os.environ['SERVICE'] = 'example'

    io_loop = ioloop.IOLoop.current()

    application = web.Application([
        (r"/", RequestHandler),
    ], **{influxdb.REQUEST_DATABASE: 'example'})
```

(continues on next page)

(continued from previous page)

```

application.listen(8888)
influxdb.install(io_loop=io_loop)
try:
    io_loop.start()
except KeyboardInterrupt:
    logging.info('Stopping')
    influxdb.shutdown()
    io_loop.stop()
    logging.info('Stopped')

```

7.1.1 Core Methods

`sprockets_influxdb.install` (*url=None, auth_username=None, auth_password=None, submission_interval=None, max_batch_size=None, max_clients=10, base_tags=None, max_buffer_size=None, trigger_size=None, sample_probability=1.0*)

Call this to install/setup the InfluxDB client collector. All arguments are optional.

Parameters

- **url** (*str*) – The InfluxDB API URL. If URL is not specified, the `INFLUXDB_SCHEME`, `INFLUXDB_HOST` and `INFLUXDB_PORT` environment variables will be used to construct the base URL. Default: `http://localhost:8086/write`
- **auth_username** (*str*) – A username to use for InfluxDB authentication. If not specified, the `INFLUXDB_USER` environment variable will be used. Default: `None`
- **auth_password** (*str*) – A password to use for InfluxDB authentication. If not specified, the `INFLUXDB_PASSWORD` environment variable will be used. Default: `None`
- **submission_interval** (*int*) – The maximum number of milliseconds to wait after the last batch submission before submitting a batch that is smaller than `trigger_size`. Default: `60000`
- **max_batch_size** (*int*) – The number of measurements to be submitted in a single HTTP request. Default: `10000`
- **max_clients** (*int*) – The number of simultaneous batch submissions that may be made at any given time. Default: `10`
- **base_tags** (*dict*) – Default tags that are to be submitted with each measurement. Default: `None`
- **max_buffer_size** (*int*) – The maximum number of pending measurements in the buffer before new measurements are discarded. Default: `25000`
- **trigger_size** (*int*) – The minimum number of measurements that are in the buffer before a batch can be submitted. Default: `5000`
- **sample_probability** (*float*) – Value between 0 and 1.0 specifying the probability that a batch will be submitted (`0.25 == 25%`)

Returns `True` if the client was installed by this call and `False` otherwise.

If `INFLUXDB_PASSWORD` is specified as an environment variable, it will be masked in the Python process.

`sprockets_influxdb.add_measurement` (*measurement*)

Add measurement data to the submission buffer for eventual writing to InfluxDB.

Example:

```
import sprockets_influxdb as influxdb

measurement = influxdb.Measurement('example', 'measurement-name')
measurement.set_tag('foo', 'bar')
measurement.set_field('baz', 1.05)

influxdb.add_measurement(measurement)
```

:param *Measurement* measurement: The measurement to add to the buffer for submission to InfluxDB.

`sprockets_influxdb.shutdown()`

Invoke on shutdown of your application to stop the periodic callbacks and flush any remaining metrics.

Returns a future that is complete when all pending metrics have been submitted.

Return type Future

7.1.2 Measurement Class

class sprockets_influxdb.**Measurement** (*database, name*)

The *Measurement* class represents what will become a single row in an InfluxDB database. Measurements are added to InfluxDB via the `add_measurement()` method.

Example:

```
import sprockets_influxdb as influxdb

measurement = Measurement('database-name', 'measurement-name')
measurement.set_tag('foo', 'bar')
measurement.set_field('baz', 1.05)

influxdb.add_measurement(measurement)
```

Parameters

- **database** (*str*) – The database name to use when submitting
- **name** (*str*) – The measurement name

duration (***kws*)

Record the time it takes to run an arbitrary code block.

Parameters **name** (*str*) – The field name to record the timing in

This method returns a context manager that records the amount of time spent inside of the context, adding the timing to the measurement.

marshall ()

Return the measurement in the line protocol format.

Return type str

set_field (*name, value*)

Set the value of a field in the measurement.

Parameters

- **name** (*str*) – The name of the field to set the value for

- **value** (*int* / *float* / *bool* / *str*) – The value of the field

Raises ValueError

set_tag (*name*, *value*)

Set a tag on the measurement.

Parameters

- **name** (*str*) – name of the tag to set
- **value** (*str*) – value to assign

This will overwrite the current value assigned to a tag if one exists.

set_tags (*tags*)

Set multiple tags for the measurement.

Parameters **tags** (*dict*) – Tag key/value pairs to assign

This will overwrite the current value assigned to a tag if one exists with the same name.

set_timestamp (*value*)

Override the timestamp of a measurement.

Parameters **value** (*float*) – The timestamp to assign to the measurement

7.1.3 Configuration Methods

`sprockets_influxdb.set_auth_credentials` (*username*, *password*)

Override the default authentication credentials obtained from the environment variable configuration.

Parameters

- **username** (*str*) – The username to use
- **password** (*str*) – The password to use

`sprockets_influxdb.set_base_url` (*url*)

Override the default base URL value created from the environment variable configuration.

Parameters **url** (*str*) – The base URL to use when submitting measurements

`sprockets_influxdb.set_max_batch_size` (*limit*)

Set a limit to the number of measurements that are submitted in a single batch that is submitted per databases.

Parameters **limit** (*int*) – The maximum number of measurements per batch

`sprockets_influxdb.set_max_buffer_size` (*limit*)

Set the maximum number of pending measurements allowed in the buffer before new measurements are discarded.

Parameters **limit** (*int*) – The maximum number of measurements per batch

`sprockets_influxdb.set_timeout` (*milliseconds*)

Override the maximum duration to wait for submitting measurements to InfluxDB.

Parameters **milliseconds** (*int*) – Maximum wait in milliseconds

`sprockets_influxdb.set_trigger_size` (*limit*)

Set the number of pending measurements that trigger the writing of data to InfluxDB

Parameters **limit** (*int*) – The minimum number of measurements to trigger a batch

7.1.4 Request Handler Mixin

class sprockets_influxdb.**InfluxDBMixin** (*application, request, **kwargs*)

Mixin that automatically submits per-request measurements to InfluxDB with the request duration.

The measurements will automatically add the following tags:

- Request handler
- Request endpoint (if enabled via a named URL)
- Request method
- Request correlation_id (if set)
- Response status_code

To add additional tags and fields, use the `set_field()`, `set_tag()`, `set_tags()`, and `timer()` methods of the `influxdb` attribute of the `RequestHandler`.

7.1.5 Other

sprockets_influxdb.**flush**()

Flush all pending measurements to InfluxDB. This will ensure that all measurements that are in the buffer for any database are written. If the requests fail, it will continue to try and submit the metrics until they are successfully written.

Return type Future

7.2 How to Contribute

Do you want to contribute fixes or improvements?

AWesome! *Thank you very much, and let's get started.*

7.2.1 Set up a development environment

The first thing that you need is a development environment so that you can run the test suite, update the documentation, and everything else that is involved in contributing. The easiest way to do that is to create a virtual environment for your endeavours:

```
$ virtualenv -p python2.7 env
```

Don't worry about writing code against previous versions of Python unless you don't have a choice. That is why we run our tests through `tox`. If you don't have a choice, then install `virtualenv` to create the environment instead. The next step is to install the development tools that this project uses. These are listed in `requires/development.txt`:

```
$ env/bin/pip install -qr requires/development.txt
```

At this point, you will have everything that you need to develop at your disposal. `setup.py` is the swiss-army knife in your development tool chest. It provides the following commands:

./setup.py nosetests Run the test suite using `nose` and generate a nice coverage report.

./setup.py build_sphinx Generate the documentation using `sphinx`.

./setup.py flake8 Run `flake8` over the code and report style violations.

If any of the preceding commands give you problems, then you will have to fix them **before** your pull request will be accepted.

7.2.2 Running Tests

The easiest (and quickest) way to run the test suite is to use the *nosetests* command. It will run the test suite against the currently installed python version and report not only the test result but the test coverage as well:

```
$ ./setup.py nosetests

running nosetests
running egg_info
writing dependency_links to sprockets-influxdb.egg-info/dependency_links.txt
writing top-level names to sprockets-influxdb.egg-info/top_level.txt
writing sprockets-influxdb.egg-info/PKG-INFO
reading manifest file 'sprockets-influxdb.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
warning: no previously-included files matching '__pycache__'...
warning: no previously-included files matching '*.swp' found ...
writing manifest file 'sprockets-influxdb.egg-info/SOURCES.txt'
...

Name                               Stmts   Miss Branch BrMiss  Cover    Missing
-----
...
TOTAL                             95      2     59      2    97%
-----
Ran 44 tests in 0.054s

OK
```

7.2.3 Submitting a Pull Request

Once you have made your modifications, gotten all of the tests to pass, and added any necessary documentation, it is time to contribute back for posterity. You've probably already cloned this repository and created a new branch. If you haven't, then checkout what you have as a branch and roll back *master* to where you found it. Then push your repository up to github and issue a pull request. Describe your changes in the request, if Travis isn't too annoyed someone will review it, and eventually merge it back.

7.3 Release History

7.3.1 2.2.0 (27 Nov 2018)

- Add support for Tornado < 6

7.3.2 2.1.0 (05 Apr 2017)

- Add sampling rate for batch submission

7.3.3 2.0.0 (09 Nov 2016)

- Change the way the buffered submission of measurements is handled

7.3.4 1.4.0 (12 Oct 2016)

- Make the hostname tag optional

7.3.5 1.3.0 (12 Oct 2016)

- Add a flag to disable submission
- Add more environment variables for configuration
- Add a maximum buffer size that discards metrics when there are too many
- Remove correlation-id fields

7.3.6 1.2.0 (23 Sep 2016)

- Make the timestamp for a measurement something that can be overridden

7.3.7 1.1.0 (23 Sep 2016)

- Submit measurements one at a time for a rejected batch, logging error responses

7.3.8 1.0.7 (14 Sep 2016)

- Have a default content length for responses without one

7.3.9 1.0.6 (14 Sep 2016)

- Move to millisecond precision

7.3.10 1.0.5 (14 Sep 2016)

- Remove `content_type` tag

7.3.11 1.0.4 (14 Sep 2016)

- Rework how the line protocol is marshalled and support the various data types.
- Remove the accept tag
- Strip down `content_type` to the `type/subtype` format only
- Make `correlation_id` a field value and not tag
- **Change the precision to second precision, per the InfluxDB docs (use the most coarse precision for better compression)**

7.3.12 1.0.3 (13 Sep 2016)

- Add a response `content_length` field, an `accept` tag (if set in request headers), and a response `content_type` tag.

7.3.13 1.0.2 (13 Sep 2016)

- Don't use `RequestHandler.reverse_url` to get the endpoint pattern in the mixin

7.3.14 1.0.1 (13 Sep 2016)

- Fixes an issue with the periodic callback

7.3.15 1.0.0 (13 Sep 2016)

- Initial release

A

`add_measurement()` (in module `sprockets_influxdb`), 17

D

`duration()` (`sprockets_influxdb.Measurement` method), 18

F

`flush()` (in module `sprockets_influxdb`), 20

I

`InfluxDBMixin` (class in `sprockets_influxdb`), 20

`install()` (in module `sprockets_influxdb`), 17

M

`marshall()` (`sprockets_influxdb.Measurement` method), 18

`Measurement` (class in `sprockets_influxdb`), 18

S

`set_auth_credentials()` (in module `sprockets_influxdb`), 19

`set_base_url()` (in module `sprockets_influxdb`), 19

`set_field()` (`sprockets_influxdb.Measurement` method), 18

`set_max_batch_size()` (in module `sprockets_influxdb`), 19

`set_max_buffer_size()` (in module `sprockets_influxdb`),
19

`set_tag()` (`sprockets_influxdb.Measurement` method), 19

`set_tags()` (`sprockets_influxdb.Measurement` method), 19

`set_timeout()` (in module `sprockets_influxdb`), 19

`set_timestamp()` (`sprockets_influxdb.Measurement`
method), 19

`set_trigger_size()` (in module `sprockets_influxdb`), 19

`shutdown()` (in module `sprockets_influxdb`), 18