
django-app-metrics Documentation

Release 0.8.0

Frank Wiles

Sep 21, 2017

Contents

1	Installation	3
1.1	Installing	3
1.2	Requirements	3
2	Usage	5
2.1	Example Code	5
2.2	Management Commands	6
3	Settings	7
3.1	Base Settings	7
3.2	Mixpanel Backend Settings	7
3.3	Statsd Settings	7
3.4	Redis Settings	8
4	Backends	9
5	Changelog	11
5.1	Version 0.8.0	11
5.2	Previous Versions	11
6	Utility Functions	13
6.1	create_metric	13
6.2	metric	13
6.3	create_metric_set	14
7	Indices and tables	15

Django App Metrics allows you to capture and report on various events in your applications.

Contents:

Installing

- Install with pip:

```
pip install django-app-metrics
```

- Add `app_metrics` to your `INSTALLED_APPS` setting:

```
INSTALLED_APPS =  
    # ...  
    'app_metrics',  
)
```

- Edit *Settings* in your project's settings module to your liking

Requirements

celery and django-celery must be installed, however if you do not wish to actually use celery you can simply set `CELERY_ALWAYS_EAGER = True` in your settings and it will behave as if celery was not configured.

Django 1.2 or higher required.

Example Code

The utility functions `create_metric` and `metric` are the main API hooks to `app_metrics`.

Example:

```
from django.contrib.auth.models import User

from app_metrics.utils import create_metric, metric

user1 = User.objects.get(pk='bob')
user2 = User.objects.get(pk='jane')

# Create a new metric to track
my_metric = create_metric(name='New User Metric', slug='new_user_signup')

# Create a MetricSet which ties a metric to an email schedule and sets
# who should receive it
my_metric_set = create_metric_set(name='My Set',
                                  metrics=[my_metric],
                                  email_recipients=[user1, user2])

# Increment the metric by one
metric('new_user_signup')

# Increment the metric by some other number
metric('new_user_signup', 4)

# Create a timer (only supported in statsd backend currently)
with timing('mytimer'):
    for x in some_long_list:
        call_time_consuming_function(x)
```

```
# Or if a context manager doesn't work for you you can use a Timer class
t = Timer()
t.start()
something_that_takes_forever()
t.stop()
t.store('mytimer')

# Gauges are current status type dials (think fuel gauge in a car)
# These simply store and retrieve a value
gauge('current_fuel', '30')
gauge('load_load', '3.14')
```

Management Commands

metrics_aggregate

Aggregate metric items into daily, weekly, monthly, and yearly totals It's fairly smart about it, so you're safe to run this as often as you like:

```
manage.py metrics_aggregate
```

metrics_send_mail

Send email reports to users. The email will be sent out using `django_mailer`'s `send_htmlmailer` if it is installed, otherwise defaults to `django.core.mail`. Can be called like:

```
manage.py metrics_send_mail
```

Base Settings

APP_METRICS_BACKEND

Defaults to `app_metrics.backends.db` if not defined.

Mixpanel Backend Settings

These settings are only necessary if you're using the *Mixpanel backend*

APP_METRICS_MIXPANEL_TOKEN

Your Mixpanel.com API token

APP_METERICS_MIXPANEL_URL

Allow overriding of the API URL end point

Statsd Settings

APP_METRICS_STATSD_HOST

Hostname of statsd server, defaults to 'localhost'

APP_METRICS_STATSD_PORT

statsd port, defaults to '8125'

APP_METRICS_STATSD_SAMPLE_RATE

statsd sample rate, defaults to 1

Redis Settings

APP_METRICS_REDIS_HOST

Hostname of redis server, defaults to 'localhost'

APP_METRICS_REDIS_PORT

redis port, defaults to '6379'

APP_METRICS_REDIS_DB

redis database number to use, defaults to 0

`app_metrics.backends.db`

This backend stores all metrics and aggregations in your database.

NOTE

Every call to `metric()` generates a database write, which may decrease your overall performance if you go nuts with them or have a heavily traffic site.

`app_metrics.backends.mixpanel`

This backend allows you to pipe all of your calls to `metric()` to Mixpanel. See the [Mixpanel documentation](#) for more information on their API.

`app_metrics.backends.statsd`

This backend allows you to pipe all of your calls to `metric()` to a statsd server. See [statsd](#) for more information on their API.

`app_metrics.backends.redis`

This backend allows you to use the `metric()` and `gauge()` aspects, but not timer aspects of `app_metrics`.

Version 0.8.0

- Added Travis CI
- Added librato backend
- Added composite backends so you can send metrics to multiple backends automatically

Previous Versions

Haven't really kept a strict history here, but we can recreate it from git logs. In short several contributors have added different backends such as statsd, redis, language translations, and docs.

API:

`create_metric`

`create_metric` (*name*, *slug*)
Creates a new type of metric track

Arguments

name The verbose name of the metric to track

slug The identifying slug used for the metric. This is what is passed into `metric()` to increment the metric

`metric`

`metric` (*slug*, *num=1*, ***kwargs*)
Increment a metric by *num*
Shortcut to the current backend (as set by `APP_METRICS_BACKEND` metric method.)

Note

If there is no metric mapped to `slug`, a metric named `Autogenerated Metric` with the passed in “`slug`” will be auto-generated.

Arguments

slug (*required*) Name of the metric to increment.

num Number to increment the metric by. Defaults to 1.

`create_metric_set`

`create_metric_set` (`create_metric_set(name=None, metrics=None, email_recipients=None, no_email=False, send_daily=True, send_weekly=False, send_monthly=False)`)
Creates a new metric set

Arguments

name Verbose name given to the new `metric_set`

metrics Iterable of slugs that the metric set should collect

email_recipients Iterable of `Users` who should be emailed with updates on the metric set

CHAPTER 7

Indices and tables

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

A

APP_METRICS_MIXPANEL_URL, 7
APP_METRICS_BACKEND, 7
APP_METRICS_MIXPANEL_TOKEN, 7
APP_METRICS_REDIS_DB, 8
APP_METRICS_REDIS_HOST, 8
APP_METRICS_REDIS_PORT, 8
APP_METRICS_STATSD_HOST, 7
APP_METRICS_STATSD_PORT, 7
APP_METRICS_STATSD_SAMPLE_RATE, 7

C

create_metric() (built-in function), 13
create_metric_set() (built-in function), 14

D

db (app_metrics.backends attribute), 9

M

metric() (built-in function), 13
mixpanel (app_metrics.backends attribute), 9

R

redis (app_metrics.backends attribute), 9

S

statsd (app_metrics.backends attribute), 9