
robust-graphite-client Documentation

Release 0.0.1

Luper Rouch

January 28, 2016

1	robgracli	3
1.1	robgracli package	3
2	Indices and tables	5
	Python Module Index	7

A simple graphite querying library with workarounds on some rare bugs.

Usage:

```
import robgracli

client = robgracli.GraphiteClient('http://my-graphite-server.com')
client.aggregate('my.awesome.metric')
print result['my.awesome.metric']
```

Reference:

1.1 robgracli package

1.1.1 Submodules

1.1.2 robgracli.client module

class `robgracli.client.GraphiteClient` (*endpoint*, *min_queries_range=600*, **args*, ***kwargs*)
Bases: `robgracli.http.HttpClient`

A simple client for querying Graphite.

Parameters

- **endpoint** – the Graphite URL;
- **min_queries_range** – The minimum range of data to query. Graphite occasionally returns empty data when querying small time ranges (probably on busy servers). The workaround is to query a larger time range and filter out unneeded values, e.g. if we want the data points from 1 minute ago, we query 10 minutes and filter out the oldest 9 minutes.

Care must be taken when choosing this value, if it's too large Graphite may return aggregated values, so it must be adapted to your storage schemas.

As a guideline, the default value of 10 minutes gave good results on our server for querying 1 minute data ranges with a `10s:1d,1min:7d,10min:1y` retention schema;

Additional arguments are passed to `robgracli.http.HttpClient`.

aggregate (*query*, *from_=60*, *aggregator=<function average>*)

Get the current value of a metric, by aggregating Graphite datapoints over an interval.

Values returned by *query* over the last *from_* seconds are aggregated using the *aggregator* function, after filtering out None values.

The return value is an `OrderedDict` with target names as keys and aggregated values as values, or None for targets that returned no datapoints or only None values.

find_metrics (*query*)

Find metrics on the server.

Querying '*' will return the root of all metrics, and you can then find other metrics from there.

Return a list of dicts of the form:

```
[
  {
    'text': 'carbon',
    'expandable': 1,
    'leaf': 0,
    'id': 'carbon',
    'allowChildren': 1
  },
  {
    'text': 'statsd',
    'expandable': 1,
    'leaf': 0,
    'id': 'statsd',
    'allowChildren': 1
  }
]
```

query (*query*, *from_=60*)

Return datapoints for *query* over the last *from_* seconds.

The return value is an `OrderedDict` with target names as keys and datapoints (value, timestamp) pairs as values.

1.1.3 robgracli.exceptions module

exception `robgracli.exceptions.BadResponse` (*response*)

Bases: `robgracli.exceptions.GraphiteException`

Raised when the graphite server returns an error response.

exception `robgracli.exceptions.GraphiteException`

Bases: `exceptions.Exception`

Base class for all errors.

1.1.4 robgracli.http module

class `robgracli.http.HttpClient` (*connect_timeout=5*, *read_timeout=5*, *max_retries=3*, *backoff_factor=1*, ***extra_requests_opts*)

Bases: `object`

A generic base class for HTTP clients, with connection pooling, sane timeouts and retries.

Parameters

- **connect_timeout** – connection timeout, in seconds;
- **read_timeout** – read timeout, in seconds;
- **max_retries** – retry requests this number of time on network errors (only works for `get()`);
- **backoff_factor** – factor used for exponential delays between retries;
- **extra_requests_opts** – additional keyword arguments passed to each `requests.Session.request()` calls.

Indices and tables

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

r

robgracli.client, 3
robgracli.exceptions, 4
robgracli.http, 4

A

aggregate() (robgracli.client.GraphiteClient method), 3

B

BadResponse, 4

F

find_metrics() (robgracli.client.GraphiteClient method), 3

G

GraphiteClient (class in robgracli.client), 3

GraphiteException, 4

H

HttpClient (class in robgracli.http), 4

Q

query() (robgracli.client.GraphiteClient method), 4

R

robgracli.client (module), 3

robgracli.exceptions (module), 4

robgracli.http (module), 4