go-metrics Documentation

Release 0.1.8

Praekelt Foundation

November 26, 2015

Contents

Contents:

CHAPTER 1

Go Metrics HTTP API

1.1 Contents

1.2 Response Format Overview

Successful responses to GET requests will contain the requested data in json format.

Example response (success response):

```
HTTP/1.1 200 OK
{
    ...
}
```

Errors are returned with the relevant HTTP error code and a json object containing `status_code`, the HTTP status code, and `reason`, the reason for the error.

Example response (error response):

```
HTTP/1.1 400 Bad Request
{
    "status_code": 400,
    "reason": "Bad Request"
}
```

1.3 Metric Types

1.3.1 Account Metrics

Account metrics are metrics relevant to a particular account, but not necessarily relevant to a particular conversation in the account. All metrics published via Vumi Go javascript sandbox applications are account metrics. Account metric names take the form `stores.<store_name>.<metric_name>.<agg_method>`:

- `store_name`: the namespace used for publishing the metrics (e.g. `default`). For javascript sandbox applications, the store name matches the configured name for the app in the conversation config unless configured otherwise.
- `metric_name`: the name of the metric (e.g. `questions_completed`).
- `agg_method`: the aggregation method used to publish metric values (e.g. `last`).

1.3.2 Conversation Metrics

Conversation metrics are metrics relevant only to a particular conversation, for example, the total messages sent in the conversation. Conversation metric names take the form `conversations.<conv_id>.<metric_name>.<agg_method>`:

- `conv_id`: the UUID for the conversation.
- `metric_name`: the name of the metric (e.g. `messages_sent`).
- `agg_method`: the aggregation method used to publish metric values (e.g. `last`).

At the time of writing, conversation metrics are only fired by internal Vumi Go processes.

1.3.3 Null Handling

The value of each datapoint returned from graphite will often be `null`. This case happens when there is no value relevant to that particular time. The api provides several ways of handling these null values:

- *zeroize*: Turns each `null` into a `0`.
- *omit*: Returns the datapoints with `null` values omitted.
- *keep*: Keeps the `null` values around.

See *`GET /api/metrics/`*'s `nulls` query parameter to see how this handling can be configured when querying the api for metrics.

1.4 API Authentication

Authentication is done using an OAuth bearer token.

Example request:

```
GET /api/metrics/ HTTP/1.1
Host: example.com
Authorization: Bearer auth-token
```

Example response (success):

```
HTTP/1.1 200 OK
```

Example response (failure):

```
HTTP/1.1 403 Forbidden
```

Example response (no authorization header):

```
HTTP/1.1 401 Unauthorized
```

1.5 API Methods

GET /api/metrics/
Retrieves the timestamp-value pairs of the metrics specified as query parameters.

Query Parameters

- **m** – Name of a metric to be retrieved. Multiple may be specified. See *Metric Types* for an overview of the metric name formats.
- **from** – The beginning time period to retrieve values from. Accepts a subset of the forms accepted by graphite. Any purely relative time (such as `-3days` or `yesterday`) is allowed, but absolute timestamps must be in the form `HH:MM_YYYYMMDD` or `YYYYMMDD`. Unix timestamp values (integer only) are also accepted. Mixing absolute and relative times is not allowed. See graphite's from and until documentation for reference, but bear these limitations in mind when reading it. Defaults to 24 hours ago.
- **until** – The ending time period to retrieve values from. Accepts a subset of the forms accepted by graphite. Any purely relative time (such as `-3days` or `yesterday`) is allowed, but absolute timestamps must be in the form `HH:MM_YYYYMMDD` or `YYYYMMDD`. Unix timestamp values (integer only) are also accepted. Mixing absolute and relative times is not allowed. See graphite's from and until documentation for reference, but bear these limitations in mind when reading it. Defaults to the current time.
- **interval** – The size of the time buckets into which metric values should be summarized. Can be in any form accepted by graphite. See graphite's functions documentation. Defaults to `1hour`.
- **align_to_from** – Align the time buckets into which metric values are summarized against to the given `from` time. Defaults to `false`.
- **nulls** – The way null `y` values returned from graphite are handled. Allowed values are `zeroize`, `omit` and `keep` (see *Null Handling*). Defaults to `zeroize`.

Example request:

```
GET /api/metrics/?m=stores.a.a.last&m=stores.b.c.avg&from=-30d&until=-1d&interval=1day&align_to_
Host: example.com
Authorization: Bearer auth-token
```

Example response (success):

```
HTTP/1.1 200 OK

{
    "stores.a.a.last": [{
      "x": 1405018164786,
      "y": 39598.0
    }, {
```

```
            "x": 1405104564786,
            "y": 36752.0
        }],
        "stores.b.c.avg": [{
            "x": 1405018164786,
            "y": 62431.0
        }, {
            "x": 1405104564786,
            "y": 72432.0
        }]
    }
```

Description of the JSON response attributes:

The response contains mappings between the metric names and an array of their timestamp-value pairs, where the pairs in the array are in ascending order of their timestamp values (from the earliest time to the latest time).

Each pair contains the timestamp under the `x` field, and is formatted as the number of milliseconds elapsed since 1 January 1970 00:00:00 UTC.

Each pair contains the value under the `y` field, and is formatted as a json number.

`POST /api/metrics/`

Fires one or many metrics as specified by the body. Body format is JSON, in the format of an object of key value pairs, where the key is the name of the metric to fire, and the value is the value to fire for the metric. Multiple metrics may be specified.

Example request:

```
POST /api/metrics/ HTTP/1.1
Host: www.example.org
Content-Type: application/json
Authorization: Bearer auth-token

{
    "metric1": 27.4,
    "metric2.sum": 11.2
}
```

Example response (success):

```
HTTP/1.1 200 OK

[
    {
        "name": "metric1",
        "value": 27.4,
        "aggregator": "avg"
    },
    {
        "name": "metric2.sum",
        "value": 11.2,
        "aggregator": "sum"
    }
]
```

Explination of aggregators:

The type of aggregator is set by the last item in the metric name, where items are separated by the fullstop `.` character. If the aggregator is not a known aggregator, or no aggregator is specified, the default aggregator of `avg` will be used.

The following is a list of aggregators that are supported by the API:

Average `avg`. Aggregates by averaging the values in each bucket.

Sum `sum`. Aggregates by summing all the values in each bucket.

Maximum `max`. Aggregates by choosing the maximum value in each bucket.

Minimum `min`. Aggregates by choosing the minimum value in each bucket.

Last `last`. Aggregates by choosing the last value in each bucket.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

/api

```
GET /api/metrics/, 5
POST /api/metrics/, 6
```