# Package ‘datadogr’

October 13, 2022

**Type**  Package  
**Title**  R Client for 'Datadog' API  
**Version**  0.1.2  
**Description**  Query for metrics from 'Datadog' (<https://www.datadoghq.com/>) via its API.  
**License**  MIT + file LICENSE  
**URL**  https://yutannihilation.github.io/K9  
**BugReports**  https://github.com/yutannihilation/K9/issues  
**Imports**  anytime, dplyr, glue, htr, lubridate, purrr, stringr, tidyr, tibble  
**Suggests**  rstudioapi, testthat, jsonlite  
**Encoding**  UTF-8  
**LazyData**  true  
**RoxygenNote**  6.0.1.9000  
**NeedsCompilation**  no  
**Author**  Hiroaki Yutani [aut, cre]  
**Maintainer**  Hiroaki Yutani <yutani.ini@gmail.com>  
**Repository**  CRAN  
**Date/Publication**  2018-05-17 16:42:07 UTC

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>datadogr</td>
<td>2</td>
</tr>
<tr>
<td>k9_auth</td>
<td>2</td>
</tr>
<tr>
<td>k9_get_events</td>
<td>2</td>
</tr>
<tr>
<td>k9_get_metrics</td>
<td>3</td>
</tr>
<tr>
<td>k9_list_metrics</td>
<td>4</td>
</tr>
<tr>
<td>k9_post_metric</td>
<td>5</td>
</tr>
</tbody>
</table>

Index 6
datadog

R Client for 'Datadog' API

Description

Query for metrics from 'Datadog' (https://www.datadoghq.com/) via its API.

k9_auth

Authentication for 'Datadog'

Description

Set API Key And Application Key

Usage

k9_auth()

k9_get_events

Get Events

Description

This end point allows you to query for event details.

Usage

k9_get_events(event_id = NULL, start = NULL, end = NULL, priority = NULL, sources = NULL, tags = NULL, .split_request = TRUE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event_id</td>
<td>Event ID.</td>
</tr>
<tr>
<td>start</td>
<td>POSIX timestamp.</td>
</tr>
<tr>
<td>end</td>
<td>POSIX timestamp.</td>
</tr>
<tr>
<td>priority</td>
<td>Priority of events. Null, &quot;low&quot; or &quot;normal&quot;.</td>
</tr>
<tr>
<td>sources</td>
<td>Sources of events. A character vector or a single comma-separated character.</td>
</tr>
<tr>
<td>tags</td>
<td>Tags of events. A named list or a single comma-separated character.</td>
</tr>
<tr>
<td>.split_request</td>
<td>if TRUE, automatically split the request when the target period is longer than a day</td>
</tr>
</tbody>
</table>
**k9_get_metrics**

*Query Time Series Points*

**Description**

This end point allows you to query for metrics from any time period.

**Usage**

```r
k9_get_metrics(query = NULL, metric = NULL, scope = NULL, by = NULL, 
from = NULL, to = NULL, .split_request = TRUE)
```

**Arguments**

- **query**
  - query string
- **metric**
  - metric name
- **scope**
  - list of scopes (scope)
- **by**
  - key to group aggregation
- **from**
  - seconds since the unix epoch
- **to**
  - seconds since the unix epoch
- **.split_request**
  - if TRUE, automatically split the request when the target period is longer than a day

**See Also**

[http://docs.datadoghq.com/api/?lang=console#events](http://docs.datadoghq.com/api/?lang=console#events)

**Examples**

```r
## Not run:
# by default get all events happened from an hour ago
k9_get_events()

# get all events happened in this week
k9_get_events(start = Sys.Date() - 7, end = Sys.Date())

# specify an event by ID
k9_get_events(event_id = "1122345566")

# specify tag
k9_get_events(tags = list(role = "db"))

## End(Not run)
```
Details

You can query either query, or the combination of metric, scope and by. For example, on the one hand you can directly query by using query = "system.cpu.idle{role:db,environment:test}by{host,region}". On the other hand, you can specify metric = "system.cpu.idle", scope = list(role = "db", environment = "test") and by = c("host", "region"), to build the same query.

Note that, if query is given, the latter will be ignored.

from and by can be one of these:

- numeric
- POSIXct
- POSIXlt
- Date
- character (parsed by anytime::anytime())
- NULL (the current epochtime will be used instead)

See Also

http://docs.datadoghq.com/api/?lang=console#metrics http://docs.datadoghq.com/graphing/

---

**k9_list_metrics**

*Get List Of Active Metrics*

Description

Get the list of actively reporting metrics from a given time until now.

Usage

k9_list_metrics(from = NULL)

Arguments

from seconds since the unix epoch

Examples

```r
## Not run:
# by default, list all metrics available since 1 hour ago
k9_list_metrics()

# if from argument is provided, this tries to get active metrics from the time
k9_list_metrics(Sys.Date() - 1)

## End(Not run)
```
k9_post_metric

**Posts a metric value to Datadog**

**Description**

This end point allows you to post time-series data that can be graphed on Datadog’s dashboards or queried from any time period.

**Usage**

\[
k9_post_metric(metric, metric_type, value, tags = list(), interval = NULL)
\]

**Arguments**

- **metric**: the name of the time series
- **metric_type**: type of your metric either: gauge, rate, or count. Optional, default=gauge
- **value**: the numeric value to post
- **tags**: a list of tags associated with the metric.
- **interval**: if the type of the metric is rate or count, define the corresponding interval. Optional, default=None

**Details**

The Datadog API uses resource-oriented URLs, uses status codes to indicate the success or failure of requests and returns JSON from all requests. With this method you can post counters, gauges to measure the value of a particular thing over time and rates that represent the derivative of a metric, it’s the value variation of a metric on a defined time interval.

**See Also**

http://docs.datadoghq.com/api/?lang=console#metrics
http://docs.datadoghq.com/graphing/
https://docs.datadoghq.com/developers/metrics/#metric-types
Index

anytime::anytime(), 4

datadogr, 2
datadogr-package (datadogr), 2

k9_auth, 2
k9_get_events, 2
k9_get_metrics, 3
k9_list_metrics, 4
k9_post_metric, 5