

# Package ‘sanapiwrapper’

May 15, 2020

**Version** 0.1

**Date** 2020-05-13

**Title** sanapiwrapper

**Description** Wrapper functions for Santiment GraphQL API.

**Author** Josef Ansinger <josef.ansinger@gmx.de>

**Maintainer** Josef Ansinger <josef.ansinger@gmx.de>

**RoxygenNote** 7.1.0

**License** GPL-3

**Imports** ghtml,  
tidyr,  
jsonlite

**Suggests** testthat

## R topics documented:

accessRestrictions . . . . .	1
availableMetrics . . . . .	2
availableSince . . . . .	2
availableSlugs . . . . .	3
santimentMetric . . . . .	3
santimentQuery . . . . .	4
setSantimentVariables . . . . .	5
<b>Index</b>	<b>6</b>

---

accessRestrictions	<i>Get access restrictions for a specific Santiment metric or query.</i>
--------------------	--

---

## Description

Get access restrictions for a specific Santiment metric or query.

## Usage

accessRestrictions(name)

**Arguments**

name                      name of Santiment metric or query

**Value**

single row with columns (isRestricted, name, restrictedFrom, restrictedTo, type)

**Examples**

```
accessRestrictions('daily_active_addresses')
```

---

availableMetrics	<i>Get the available Santiment metrics for a specific project.</i>
------------------	--

---

**Description**

Returns all available timeseries metrics (note not the histogram metrics) for a slug.

**Usage**

```
availableMetrics(slug)
```

**Arguments**

slug                      project

**Value**

vector of available metrics

**Examples**

```
availableMetrics('ethereum')
```

---

availableSince	<i>Get the earliest date for which the Santiment metric is available.</i>
----------------	---

---

**Description**

Get the earliest date for which the Santiment metric is available.

**Usage**

```
availableSince(metric, slug)
```

**Arguments**

metric                      metric  
slug                          project

**Value**

earliest date

**Examples**

```
availableSince('daily_active_addresses', 'ethereum')
```

---

availableSlugs

*Get the available slugs for a specific metric*

---

**Description**

Get the available slugs for a specific metric

**Usage**

```
availableSlugs(metric)
```

**Arguments**

metric                      Santiment metric

**Value**

vector of available slugs

**Examples**

```
availableSlugs('daily_active_addresses')
```

---

santimentMetric

*Get a metric from the Santiment GraphQL API.*

---

**Description**

Checks if the metric is available for the slug. If not, a constant vector is returned. Checks if the metric is available for this time window. For unavailable dates, zeros are returned.

**Usage**

```
santimentMetric(  
  metric,  
  slug,  
  from = "2019-01-01",  
  to = "2020-01-01",  
  aggregation = "SUM",  
  selector_option = NULL  
)
```

**Arguments**

metric	Santiment metric
slug	project (e.g. ethereum, bitcoin, tezos ...)
from	start of the time window
to	end of the time window
aggregation	aggregation (SUM, AVG, MEDIAN, MIN, MAX)
selector_option	optional string for selector field, for example "holdersCount:10" for "amount_in_top_holders" metric, see "selector: slug: "ethereum", holdersCount:10"

**Details**

<https://neuro.santiment.net> <https://graphql.org>

**Value**

table with columns (date, <metric>)

**Examples**

```
santimentMetric('dev_activity', 'ethereum')
```

---

santimentQuery	<i>Execute Santiment query.</i>
----------------	---------------------------------

---

**Description**

A new GraphQL client is generated. A new query is generated, filled with the query string and (optional) variables, and then executed.

**Usage**

```
santimentQuery(query_string, query_variables)
```

**Arguments**

query_string	query string
query_variables	optional list of query variables

**Value**

query result

**Examples**

```
sentimentQuery('{projectBySlug(slug: "ethereum") {availableQueries}}')

string <- 'query MyQuery($slug: String = \"bitcoin\")
  {projectBySlug(slug: $slug) {availableQueries}}'
variables <- list(slug = 'ethereum')
sentimentQuery(string, variables)
```

---

setSentimentVariables *Set API key and API URL for Santiment GraphQL queries.*

---

**Description**

Note that setting API key and API URL is optional. Without them queries will default to a free Santiment account.

**Usage**

```
setSentimentVariables(key, url = "https://api.santiment.net/graphql")
```

**Arguments**

key	API key
url	GraphQL server URL

**Examples**

```
setSentimentVariables("y4i3n5xk6p")
```

# Index

accessRestrictions, [1](#)  
availableMetrics, [2](#)  
availableSince, [2](#)  
availableSlugs, [3](#)  
  
sentimentMetric, [3](#)  
sentimentQuery, [4](#)  
setSentimentVariables, [5](#)