

# Package ‘piwikproR’

October 14, 2022

**Type** Package

**Title** Access 'Piwik Pro' Website Statistics

**Date** 2022-08-29

**Version** 0.4.0

**Author** Martin Stingl <martin.stingl@dfv.de>

**Maintainer** Martin Stingl <martin.stingl@dfv.de>

**Description** Run Queries against the API of 'Piwik Pro' <[https://developers.piwik.pro/en/latest/custom\\_reports/http\\_api/http\\_api.html](https://developers.piwik.pro/en/latest/custom_reports/http_api/http_api.html)>. The result is a tibble.

**URL** <https://piwikproR.rstats-tips.net>

**BugReports** <https://github.com/dfv-ms/piwikproR/issues>

**License** GPL (>= 3)

**Encoding** UTF-8

**Imports** httr, rjson, tibble, purrr, lubridate, magrittr, dplyr, rlang, stringr, readr (>= 2.1.0), digest, fs, tidyr, glue

**RoxygenNote** 7.2.1

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-09-01 09:10:38 UTC

## R topics documented:

|                          |   |
|--------------------------|---|
| apply_types . . . . .    | 2 |
| build_filter . . . . .   | 3 |
| build_query . . . . .    | 4 |
| get_apps_list . . . . .  | 5 |
| get_app_detail . . . . . | 5 |

|   |    |
|---|----|
| get_column_type . . . . .                     | 6  |
| get_generic_list . . . . .                    | 6  |
| get_login_token . . . . .                     | 7  |
| get_metasites_list . . . . .                  | 7  |
| get_permissions_for_app . . . . .             | 8  |
| get_permissions_for_metasite . . . . .        | 8  |
| get_test_credentials . . . . .                | 9  |
| get_usergroups_list . . . . .                 | 9  |
| get_usergroup_members . . . . .               | 10 |
| get_users_list . . . . .                      | 10 |
| is_column_a_metric . . . . .                  | 11 |
| MAX_LINES_PER_REQUEST_ANALYTICS_API . . . . . | 11 |
| MAX_LINES_PER_REQUEST_USERS_API . . . . .     | 12 |
| release_questions . . . . .                   | 12 |
| send_query . . . . .                          | 12 |
| send_query_single . . . . .                   | 13 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>15</b> |
|--------------|-----------|

---

|             |   |
|-------------|---|
| apply_types | <i>Convert column-type according to column_name</i> |
|-------------|---|

---

## Description

Convert column-type according to column\_name

## Usage

```
apply_types(data, timestamp_to_date = TRUE)
```

## Arguments

|                   |                                   |
|-------------------|-----------------------------------|
| data              | tibble                            |
| timestamp_to_date | boolean convert timestamp to date |

## Value

tibble

---

|              |                                  |
|--------------|----------------------------------|
| build_filter | <i>Build filter from tribble</i> |
|--------------|----------------------------------|

---

**Description**

Build filter from tribble

**Usage**

```
build_filter(filters, global_operator = "and")
```

**Arguments**

|                 |   |
|-----------------|---|
| filters         | tribble with columns <ul style="list-style-type: none"><li>• column</li><li>• operator</li><li>• value</li></ul> Possible values for operator: <ul style="list-style-type: none"><li>• (not_)contains</li><li>• (not_)icontains</li><li>• start_with</li><li>• ends_with</li><li>• (not_)matches</li><li>• eq</li><li>• neq</li><li>• (not_)empty</li></ul> |
| global_operator | "and" or "or"   |

**Value**

filter json encoded to feed to [build\\_query](#)

**Examples**

```
filters <- tibble::tribble(
  ~column, ~operator, ~value,
  "event_url", "matches", "Zamperoni",
  "event_url", "matches", "-1[34]"
)
# With optional transformation
filters <- tibble::tribble(
  ~column, ~operator, ~value, ~transformation,
  "event_url", "starts_with", "/medien", "to_path",
  "event_url", "matches", "-1[34]", NULL
)
```

```

filters <- build_filter(filters, "and")

# If values of two or more different types are used use lists

filters <- tibble::tribble(
  ~column, ~operator, ~value,
  "device_type", "eq", list(0),
  "location_country_name", "eq", list("DE")
)

# This also works for transformation

```

---

 build\_query

*Build the query*


---

### Description

Build the query

### Usage

```

build_query(
  date_from,
  date_to,
  website_id,
  columns,
  filters = NULL,
  metric_filters = NULL,
  offset = 0,
  max_lines = 0
)

```

### Arguments

|                |  |
|----------------|--|
| date_from      | Start date of query  |
| date_to        | End date of query  |
| website_id     | website_id from piwik  |
| columns        | tibble containing columns and transformations (metrics and dimensions) |
| filters        | list containing filter, best built by <a href="#">build_filter</a>     |
| metric_filters | list containing filter, best built by <a href="#">build_filter</a>     |
| offset         | offset   |
| max_lines      | limit  |

### Value

query as list

**Examples**

```
columns <- tibble::tribble(
  ~column, ~transformation,
  "event_url", "to_path",
  "event_url", "to_domain",
  "website_name", "",
  "timestamp", "",
  "timestamp", "to_hour_of_day",
  "page_views", ""
)
build_query(lubridate::ymd("2021-01-01"), lubridate::ymd("2021-01-19"),
  "xxx", columns)
```

---

|               |                                     |
|---------------|-------------------------------------|
| get_apps_list | <i>get_apps_list fetch all apps</i> |
|---------------|-------------------------------------|

---

**Description**

get\_apps\_list fetch all apps

**Usage**

```
get_apps_list(token)
```

**Arguments**

token            login token generated by [get\\_login\\_token](#)

**Value**

data.frame

---

|                |   |
|----------------|---|
| get_app_detail | <i>get_app_detail fetching details about an app</i> |
|----------------|---|

---

**Description**

get\_app\_detail fetching details about an app

**Usage**

```
get_app_detail(token, app)
```

**Arguments**

token            login token generated by [get\\_login\\_token](#)  
 app              app id, see [get\\_apps\\_list](#)

**Value**

data.frame with details about the app

---

|                 |   |
|-----------------|---|
| get_column_type | <i>Convert column-type according to column_name</i> |
|-----------------|---|

---

**Description**

Convert column-type according to column\_name

**Usage**

```
get_column_type(column_name, timestamp_to_date = TRUE)
```

**Arguments**

column\_name    string  
 timestamp\_to\_date    boolean convert timestamp to date

**Value**

string suggested type of column

---

|                  |   |
|------------------|---|
| get_generic_list | <i>get_generic_list fetches all rows of a given api</i> |
|------------------|---|

---

**Description**

get\_generic\_list fetches all rows of a given api

**Usage**

```
get_generic_list(token, offset, api_endpoint, columns, datetime_columns)
```

**Arguments**

- token                login token generated by [get\\_login\\_token](#)
- offset              default 0, integer, where to start fetching usergroups
- api\_endpoint       string with api endpoint
- columns             vector of strings containing columnnames
- datetime\_columns   vector of strings containing columnnames which are converted to datetime

**Value**

data.frame

---

|                              |                          |
|------------------------------|--------------------------|
| <code>get_login_token</code> | <i>Fetch login token</i> |
|------------------------------|--------------------------|

---

**Description**

Fetch login token

**Usage**

`get_login_token(credentials)`

**Arguments**

credentials        List with fields client\_id, client\_secret and url

**Value**

List with login\_token

---

|                                 |                           |
|---------------------------------|---------------------------|
| <code>get_metasites_list</code> | <i>get_metasites_list</i> |
|---------------------------------|---------------------------|

---

**Description**

get\_metasites\_list

**Usage**

`get_metasites_list(token)`

**Arguments**

token                login token generated by [get\\_login\\_token](#)

**Value**

data.frame

---

`get_permissions_for_app`  
*get\_permissions\_for\_app*

---

**Description**

`get_permissions_for_app`

**Usage**

```
get_permissions_for_app(token, app)
```

**Arguments**

|       |  |
|-------|--|
| token | login token generated by <a href="#">get_login_token</a> |
| app   | app id, see <a href="#">get_apps_list</a>                |

**Value**

data.frame

---

`get_permissions_for_metasite`  
*get\_permissions\_for\_metasite*

---

**Description**

`get_permissions_for_metasite`

**Usage**

```
get_permissions_for_metasite(token, metasite)
```

**Arguments**

|          |  |
|----------|--|
| token    | login token generated by <a href="#">get_login_token</a> |
| metasite | app id, see <a href="#">get_metasites_list</a>           |

**Value**

data.frame



---

`get_test_credentials` *get\_test\_credentials getting credentials for testing*

---

### **Description**

Fills credentials out of ENV into a list

### **Usage**

```
get_test_credentials()
```

### **Value**

list

---

`get_usergroups_list` *get\_usergroups\_list fetch all usergroups*

---

### **Description**

`get_usergroups_list` fetch all usergroups

### **Usage**

```
get_usergroups_list(token)
```

### **Arguments**

token            login token generated by [get\\_login\\_token](#)

### **Value**

data.frame

---

`get_usergroup_members` *get\_usergroup\_members*

---

**Description**

`get_usergroup_members`

**Usage**

`get_usergroup_members(token, usergroup_id)`

**Arguments**

`token` login token generated by [get\\_login\\_token](#)  
`usergroup_id` usergroup (see [get\\_usergroups\\_list](#))

**Value**

`data.frame`

---

`get_users_list` *get\_users\_list fetch all users*

---

**Description**

`get_users_list` fetch all users

**Usage**

`get_users_list(token)`

**Arguments**

`token` login token generated by [get\\_login\\_token](#)

**Value**

`data.frame`

---

|                    |                           |                  |                |  |
|--------------------|---------------------------|------------------|----------------|--|
| is_column_a_metric | <i>is_column_a_metric</i> | <i>Checks</i>    | <i>if</i>      | <i>col-</i>  |
|                    | <i>umn_name</i>           | <i>indicates</i> | <i>numeric</i> | <i>values</i>  |
|                    |                           |                  |                | <i>Uses</i>  |
|                    |                           |                  |                | <i><a href="https://developers.piwik.pro/en/latest/custom_reports/columns.html">https://developers.piwik.pro/en/latest/custom_reports/columns.html</a></i> |

---

**Description**

is\_column\_a\_metric Checks if column\_name indicates numeric values Uses [https://developers.piwik.pro/en/latest/custom\\_reports/columns.html](https://developers.piwik.pro/en/latest/custom_reports/columns.html)

**Usage**

```
is_column_a_metric(column_name)
```

**Arguments**

column\_name    string

**Value**

boolean

---

MAX\_LINES\_PER\_REQUEST\_ANALYTICS\_API

*Maximum number of lines requested in Analytics-API*

---

**Description**

Maximum number of lines requested in Analytics-API

**Usage**

```
MAX_LINES_PER_REQUEST_ANALYTICS_API()
```

**Value**

int

---

MAX\_LINES\_PER\_REQUEST\_USERS\_API

*Maximum number of lines requested in Users-API*

---

**Description**

Maximum number of lines requested in Users-API

**Usage**

MAX\_LINES\_PER\_REQUEST\_USERS\_API()

**Value**

int

---

release\_questions

*Ask package specific question during release-process*

---

**Description**

Ask package specific question during release-process

**Usage**

release\_questions()

**Value**

vector of strings

---

send\_query

*Send the query and receive the result*

---

**Description**

Send the query and receive the result

**Usage**

```
send_query(
  query,
  token,
  use_csv = TRUE,
  fetch_by_day = FALSE,
  api = "query",
  caching = FALSE,
  caching_dir = "cache",
  convert_types = TRUE
)
```

**Arguments**

|               |  |
|---------------|--|
| query         | list generated by <a href="#">build_query</a>                  |
| token         | login token generated by <a href="#">get_login_token</a>       |
| use_csv       | logical to choose whether to fetch data via extra csv-request  |
| fetch_by_day  | logical fetch data day by day                                  |
| api           | API endpoint ("query", "sessions", "events")                   |
| caching       | logical Set TRUE to enable caching                             |
| caching_dir   | character Set directory for saving caching data, default cache |
| convert_types | logical guess type of columns and set them                     |

**Value**

result as tibble

---

|                   |  |
|-------------------|--|
| send_query_single | <i>Send the query and receive the result</i> |
|-------------------|--|

---

**Description**

Send the query and receive the result

**Usage**

```
send_query_single(query, token, use_csv, api, caching, caching_dir)
```

**Arguments**

|             |   |
|-------------|---|
| query       | list generated by <a href="#">build_query()</a>               |
| token       | login token   |
| use_csv     | logical to choose whether to fetch data via extra csv-request |
| api         | API endpoint (query, sessions, events)                        |
| caching     | logical Set TRUE to enable caching                            |
| caching_dir | character Set directory for saving caching data               |

**Value**

result as list with values data and meta

# Index

[apply\\_types](#), [2](#)

[build\\_filter](#), [3](#), [4](#)

[build\\_query](#), [3](#), [4](#), [13](#)

[get\\_app\\_detail](#), [5](#)

[get\\_apps\\_list](#), [5](#), [6](#), [8](#)

[get\\_column\\_type](#), [6](#)

[get\\_generic\\_list](#), [6](#)

[get\\_login\\_token](#), [5-7](#), [7](#), [8-10](#), [13](#)

[get\\_metasites\\_list](#), [7](#), [8](#)

[get\\_permissions\\_for\\_app](#), [8](#)

[get\\_permissions\\_for\\_metasite](#), [8](#)

[get\\_test\\_credentials](#), [9](#)

[get\\_usergroup\\_members](#), [10](#)

[get\\_usergroups\\_list](#), [9](#), [10](#)

[get\\_users\\_list](#), [10](#)

[is\\_column\\_a\\_metric](#), [11](#)

[MAX\\_LINES\\_PER\\_REQUEST\\_ANALYTICS\\_API](#),  
[11](#)

[MAX\\_LINES\\_PER\\_REQUEST\\_USERS\\_API](#), [12](#)

[release\\_questions](#), [12](#)

[send\\_query](#), [12](#)

[send\\_query\\_single](#), [13](#)