

Package ‘rym’

June 20, 2020

Type Package

Title R Interface to Yandex Metrica API

Version 1.0.0

Date 2020-06-11

Author Alexey Seleznev <selesnow@gmail.com>

Maintainer Alexey Seleznev <selesnow@gmail.com>

Description Allows work with 'Management API' for load counters, segments, filters, user permissions and goals list from Yandex Metrica, 'Reporting API' allows you to get information about the statistics of site visits and other data without using the web interface, 'Logs API' allows to receive non-aggregated data and 'Compatible with Google Analytics Core Reporting API v3' allows receive information about site traffic and other data using field names from Google Analytics Core API. For more information see official documents <<https://tech.yandex.ru/metrika/doc/api2/concept/about-docpage/>>.

Depends R (>= 3.5.0)

BugReports <https://github.com/selesnow/rym/issues>

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL-2

Imports httr, stringr, utils, purrr

URL <http://selesnow.github.io/rym>

Encoding UTF-8

Language ru

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-20 08:40:03 UTC

R topics documented:

rym-package	2
rym-calls	4
rym-expense-uploading	7
rym-offline-conversion	9
rym_add_goal	11
rym_add_segment	12
rym_auth	13
rym_get_counters	14
rym_get_data	15
rym_get_direct_clients	17
rym_get_filters	18
rym_get_ga	19
rym_get_goals	20
rym_get_logs	21
rym_get_my_logins	22
rym_get_segments	23
rym_users_grants	24
Index	26

 rym-package

R Interface to Yandex Metrika API

Description

Allows work with 'Management API' for load counters, segments, filters, user permissions and goals list from Yandex Metrika, 'Reporting API' allows you to get information about the statistics of site visits and other data without using the web interface, 'Logs API' allows to receive non-aggregated data and 'Compatible with Google Analytics Core Reporting API v3' allows receive information about site traffic and other data using field names from Google Analytics Core API. For more information see official documents <<https://tech.yandex.ru/metrika/doc/api2/concept/about-docpage/>>.

Details

The DESCRIPTION file:

```

Package:      rym
Type:         Package
Title:        R Interface to Yandex Metrika API
Version:      1.0.0
Date:         2020-06-11
Author:       Alexey Seleznev <selesnow@gmail.com>
Maintainer:  Alexey Seleznev <selesnow@gmail.com>
Description:  Allows work with 'Management API' for load counters, segments, filters, user permissions and goals list fr
Depends:      R (>= 3.5.0)

```

BugReports: <https://github.com/selesnow/rym/issues>
 Suggests: knitr, rmarkdown
 VignetteBuilder: knitr
 License: GPL-2
 Imports: httr, stringr, utils, purrr
 URL: <http://selesnow.github.io/rym>
 Encoding: UTF-8
 Language: ru

Index of help topics:

rym-calls	Manage callss
rym-expense-uploading	Manage Cost Data Upload
rym-offline-conversion	Manage offline conversions
rym-package	R Interface to Yandex Metrika API
rym_add_goal	Create goal in 'Yandex Metrika'
rym_add_segment	Create segment in 'Yandex Metrika'
rym_auth	Authentication in Yandex Metrika API
rym_get_counters	Load Yandex Metrika counters
rym_get_data	Load data by compatible with the 'Google Analytics Core Reporting API'
rym_get_direct_clients	Load Yandex.Direct clients
rym_get_filters	List of filters
rym_get_ga	Work with 'compatible with the Google Analytics Core Reporting API (v3)'
rym_get_goals	List of goals
rym_get_logs	Get raw data from yandex metrika.
rym_get_my_logins	Show list of auth logins
rym_get_segments	List of segments
rym_users_grants	List of users permissions

Author(s)

Alexey Seleznev <selesnow@gmail.com>
 Maintainer: Alexey Seleznev <selesnow@gmail.com>

References

[Officialrym documents](#)
[Management API Documents](#)
[Reporting API Documents](#)
[API Compatible with Google Analytics Core API v3](#)
[Logs API Documentations](#)
 Or read vignettes:

```
vignette("intro-to-rym", package = "rym")
vignette("rym-management-api", package = "rym")
vignette("rym-reporting-api", package = "rym")
vignette("rym-ga-api", package = "rym")
vignette("rym-logs-api", package = "rym")
```

Examples

```
## Not run:
library(rym)

# get counters list
my_counters <- rym_get_counter()

# join all counters id in string format, like 1,2,3,4,5,...,n
counters_ids <- paste0(my_counters$id, collapse = ",")

# get statistic from reporting API
reporting <- rym_get_data(counters = counters_ids,
                          date.from = "2018-08-01",
                          date.to = "yesterday",
                          dimensions = "ym:s:date,
ym:s:lastTrafficSource",
                          metrics = "ym:s:visits,
ym:s:pageviews,
ym:s:users",
                          sort = "-ym:s:date")

# get raw data
raw_data <- rym_get_logs(counter = counters_ids[1],
                          date.from = "2016-12-01",
                          date.to = "2016-12-20",
                          fields = "ym:s:visitID,
ym:s:date,
ym:s:bounce,
ym:s:clientID,
ym:s:networkType",
                          source = "visits")

## End(Not run)
```

rym-calls

Manage callss

Description

calls Management:

rym_enable_calls Enable calls tracking for the specified counter.

rym_disable_calls Disable calls tracking for the specified counter.

rym_upload_calls Upload callss.

rym_allow_calls Returns the date and time from which conversions can be linked to visits for the specified counter.

rym_get_uploadings_calls Returns a list of calls downloads.

Usage

```
rym_enable_calls(
  counter,
  login      = getOption("rym.user"),
  token.path = getOption("rym.token_path"))

rym_disable_calls(
  counter,
  login      = getOption("rym.user"),
  token.path = getOption("rym.token_path"))

rym_allow_calls(
  counter,
  login      = getOption("rym.user"),
  token.path = getOption("rym.token_path"))

rym_upload_calls(
  counter,
  data,
  client.id.type = c("CLIENT_ID", "USER_ID"),
  new.goal.name  = NULL,
  comment        = paste0("Upload by rym at ", Sys.time()),
  login          = getOption("rym.user"),
  token.path     = getOption("rym.token_path"))

rym_get_uploadings_calls(
  counter,
  login      = getOption("rym.user"),
  token.path = getOption("rym.token_path"))
```

Arguments

counter	Your yandex metrika counter
data	data.frame, expense data for loading
client.id.type	Type of visitor identifiers: CLIENT_ID or USER_ID
comment	Comment for data loading
new.goal.name	The name of the new goal for calls in the event that such a goal does not exist. If the target exists, this parameter is ignored
login	character, Your Yandex login
token.path	character, Directory for store API credential

Details

Loading data format.

Require:

UserId, ClientId, Yclid Identifier of the visitor to the site or advertising campaign in Yandex.Direct, for example: 12345

Optional:

StaticCall whether the call is static (1 - static, 0 - dynamic)

Price target price, the decimal separator is the dot (.)

Currency currency in the three-letter format ISO 4217, for example: RUB

PhoneNumber phone number without spaces (with country and city code). For example, +70123456789

TalkDuration call duration in seconds

HoldDuration call waiting time in seconds

CallMissed whether the call is missed (1 - missed, 0 - answered)

Tag custom label. It can be used to mark the quality of a call or its outcome, etc. For example, “the customer was not satisfied with the price”

FirstTimeCaller primary (1 - primary call, 0 - secondary call)

URL The URL from which the call came (the page associated with the event). For example, this could be the landing page for an ad campaign that has a phone number (PhoneNumber)

CallTrackerURL The URL to go to the call tracker interface

Example of [CSV file](#).

Author(s)

Alexey Seleznev

References

See official dox: [API documents Upload calls info into Yandex Metrica Calls in Yandex Metrica](#)

Examples

```
## Not run:
# read data for upload
data <- read.csv("https://bit.ly/2CcsNyl")

# enable offline cnversion
rym_enable_calls(
  counter = 123456789,
  login   = "your_login"
)

# upload data
rym_upload_calls(
  counter = 123456789,
```

```
data          = data,
client.id.type = "CLIENT_ID",
login         = "your_login")

# check uploading
rym_allow_calls(
  counter = 123456789,
  login   = "your_login")

## End(Not run)
```

rym-expense-uploading *Manage Cost Data Upload*

Description

Upload, delete and get list of uploading expense data in 'Yandex Metrica'.

Usage

```
rym_upload_expense(
  counter,
  data,
  comment = paste0("Upload bt rym at ", Sys.time()),
  login   = getOption("rym.user"),
  token.path = getOption("rym.token_path"))

rym_delete_uploaded_expense(
  counter,
  data,
  comment = paste0("Upload bt rym at ", Sys.time()),
  login   = getOption("rym.user"),
  token.path = getOption("rym.token_path"))

rym_get_uploadings_expense(
  counter,
  login   = getOption("rym.user"),
  token.path = getOption("rym.token_path"))
```

Arguments

counter	Your yandex metrica counter
data	data.frame, expense data for loading
comment	Comment for data loading
login	character, Your Yandex login
token.path	character, Directory for store API credential

Details

Loading data format.

Require:

Date Date in YYYY-MM-DD format

UTMSource Source name, for example google

Expenses Expense size, for example 12.34

Optional:

UTMMedium Medium name, for example cpc

UTMCampaign Campaign name, for example my campaign

UTMTerm Term name, for example my term

UTMContent Content name, for example my content

Currency Currency in [ISO 4217](#) code

Clicks Clicks number, for example 23

Author(s)

Alexey Seleznev

References

See official docs: [API documents](#) [Yandex Metrika documents](#)

Examples

```
## Not run:
# prepare data for loading
expense <- data.frame(Date       = c("2020-06-01",
                                     "2020-06-02"),
                      UTMSource  = c("test_s_1",
                                     "test_s_2"),
                      Expenses   = c(88.12,
                                     92.11),
                      UTMMedium  = c("cpc",
                                     "cpm"),
                      UTMCampaign = c("camp1",
                                     "camp2"),
                      UTMTerm    = c("term1",
                                     "term2"),
                      UTMContent  = c("cont1",
                                     "cont2"),
                      Currency    = c("RUB",
                                     "RUB"),
                      Clicks      = c(11, 15))

# upload data
rym_upload_expense(
```



```

        counter = 1111111,
        data     = expense,
        login    = 'yandex_login')

# delete data
rym_delete_uploaded_expense(
    counter = 1111111,
    data     = expense,
    login    = 'yandex_login')

# get list of expense uploadings
loaded <- rym_get_uploadings_expense(
    counter = 1111111,
    login    = 'yandex_login')

## End(Not run)

```

rym-offline-conversion

Manage offline conversions

Description

Offline Conversion Management:

rym_enable_offline_conversion Enable offline conversion tracking for the specified counter.

rym_disable_offline_conversion Disable offline conversion tracking for the specified counter.

rym_upload_offline_conversion Upload offline conversions.

rym_allow_offline_conversion Returns the date and time from which conversions can be linked to visits for the specified counter.

rym_get_uploadings_offline_conversions Returns a list of offline conversion downloads.

Usage

```

rym_enable_offline_conversion(
  counter,
  login    = getOption("rym.user"),
  token.path = getOption("rym.token_path"))

rym_disable_offline_conversion(
  counter,
  login    = getOption("rym.user"),
  token.path = getOption("rym.token_path"))

rym_allow_offline_conversion(
  counter,

```

```

login      = getOption("rym.user"),
token.path = getOption("rym.token_path"))

rym_upload_offline_conversion(
  counter,
  data,
  client.id.type = c("CLIENT_ID", "USER_ID"),
  comment       = paste0("Upload by rym at ", Sys.time()),
  login        = getOption("rym.user"),
  token.path    = getOption("rym.token_path"))

rym_get_uploadings_offline_conversions(
  counter,
  login      = getOption("rym.user"),
  token.path = getOption("rym.token_path"))

```

Arguments

counter	Your yandex metrika counter
data	data.frame, expense data for loading
client.id.type	Type of visitor identifiers: CLIENT_ID or USER_ID
comment	Comment for data loading
login	character, Your Yandex login
token.path	character, Directory for store API credential

Details

Loading data format.

Require:

UserId, ClientId, Yclid Identifier of the visitor to the site or advertising campaign in Yandex.Direct, for example: 12345

Target Target ID specified when creating the target, for example order_confirmed

DateTime The time at which the conversion occurred in Unix Time Stamp format. Indicate the time in the time zone UTC + 0, for example: 1481718166

Optional:

Price Goal Value, for example: 1000

Currency Three-letter currency code according to ISO 4217, for example: RUB

Example of [CSV file](#).

Author(s)

Alexey Seleznev

References

See official dox: [API documents](#) [Yandex Metrica documents](#)

Examples

```
## Not run:
# read data for upload
data <- read.csv("https://bit.ly/2CcsNyl")

# enable offline cnversion
rym_enable_offline_conversion(
  counter = 123456789,
  login   = "your_login"
)

# upload data
rym_upload_offline_conversion(
  counter      = 123456789,
  data         = data,
  client.id.type = "CLIENT_ID",
  login        = "your_login")

# check uploading
rym_allow_offline_conversion(
  counter = 123456789,
  login   = "your_login")

## End(Not run)
```

rym_add_goal

Create goal in 'Yandex Metrica'

Description

Create new goal in 'Yandex Metrica' counter.

Usage

```
rym_add_goal(
  counter,
  name,
  type = c("number", "action", "step", "url"),
  is.retargeting = FALSE,
  flag = c(NA, "basket", "order"),
  conditions = NULL,
  login = getOption("rym.user"),
  token.path = getOption("rym.token_path"))
```

Arguments

counter	Yandex Metrika counter ID
name	character, goal name
type	character, goal type, one of "number","action","step","url"
is.retargeting	boolean, if TRUE the goal is retargeting
flag	character, target type for Yandex.Market customers
conditions	list, the list of conditions, for example <code>list(type = 'exact',url = 'rym-first-goal')</code>
login	character, Your Yandex login
token.path	character, Directory for store API credential

Author(s)

Alexey Seleznev

References

[Official docs of 'Yandex Metrika Management API'](#)

Examples

```
## Not run:
rym_add_goal(123456789,
             name = 'first_goal',
             type = 'action',
             conditions = list(type = 'exact',
                               url = 'rym-first-goal'),
             login = 'your_login')

## End(Not run)
```

rym_add_segment

Create segment in 'Yandex Metrika'

Description

Create a new API segment in 'Yandex Metrika' counter.

Usage

```
rym_add_segment(
  counter,
  name,
  expression,
  login = getOption("rym.user"),
  token.path = getOption("rym.token_path"))
```

Arguments

counter	Yandex Metrika counter ID
name	character, segment name
expression	character, filter expression, for example "ym:s:trafficSource=='organic' AND ym:s:isNewUser=='Yes' ", for more information go link
login	character, Your Yandex login
token.path	character, Directory for store API credential

Author(s)

Alexey Seleznev

References

[Official docs of 'Yandex Metrika Management API'](#)

See Also

See [rym_add_goal\(\)](#) for create new goal in 'Yandex Metrika'.

Examples

```
## Not run:
rym_add_segment(
  counter = 123456789,
  name = "my_segment",
  expression = "ym:s:trafficSource=='organic' AND ym:s:isNewUser=='Yes'",
  login = "your_login")

## End(Not run)
```

rym_auth

Authentication in Yandex Metrika API

Description

Actually you don't need call rym_auth since it will be called when you run any of the functions available in rym.

Usage

```
rym_auth(login = getOption("rym.user"),
         new.user = FALSE,
         token.path = getOption("rym.token_path"))
```

Arguments

login	character, Your Yandex login
new.user	logical, If TRUE you reauth in Yandex Metrika API
token.path	character, Directory for store API credential

Details

All your credential save in directory specified in the argument token.path, after save every time you run any function from rym, rym_auth be load credentials from local file login.rymAuth.RData. rym_auth create files for each your logins, which makes it possible use many yandex account in one script. Every tokens expire after 1 year after authentication, but rym_auth automaticly refresh and save you credential 30 days before he expire.

Value

List with credential data.

Author(s)

Alexey Seleznev

References

OAuth 2.0 in Yandex docs - <https://tech.yandex.ru/oauth/doc/dg/concepts/about-docpage/>

Examples

```
## Not run:  
library(rym)  
rym_auth(login = "my_login")  
  
## End(Not run)
```

rym_get_counters	<i>Load Yandex Metrika counters</i>
------------------	-------------------------------------

Description

Returns a list of existing counters available to the user.

Usage

```
rym_get_counters(login = getOption("rym.user"),  
                 token.path = getOption("rym.token_path"),  
                 search.string = NULL)
```

Arguments

login	character, Your Yandex login
token.path	character, Directory for store API credential
search.string	character, Filter by line. Counters will be displayed, the identifier, name, site or mirrors of which contain the specified substring

Value

Data frame with list of yandex metrika counters.

Author(s)

Alexey Seleznev

References

Official docs - <https://tech.yandex.ru/metrika/doc/api2/management/counters/counters-docpage/>

Examples

```
## Not run:  
my_counters <- rym_get_counters(login = "my_login")  
  
## End(Not run)
```

rym_get_data	<i>Load data by compatible with the 'Google Analytics Core Reporting API'</i>
--------------	---

Description

The Reporting API allows you to obtain information about site visits statistics and other data without using the Yandex.Metrica interface.

Usage

```
rym_get_data(direct.client.logins = NULL, counters,  
  metrics = "ym:s:visits,ym:s:pageviews,ym:s:users",  
  dimensions = NULL, filters = NULL,  
  sort = NULL, date.from = "8daysAgo",  
  date.to = "yesterday", accuracy = "full",  
  include.undefined = TRUE, lang = "ru",  
  timezone = NULL, pretty = FALSE,  
  login = getOption("rym.user"),  
  token.path = getOption("rym.token_path"))
```

Arguments

<code>direct.client.logins</code>	Logins of Yandex.Direct clients, separated by commas. Can be used to generate a Direct Expense report.
<code>counters</code>	Counter identifiers, separated by commas.
<code>metrics</code>	A list of metrics, separated by a comma.
<code>dimensions</code>	A list of groupings, separated by a comma.
<code>filters</code>	A list of filtering data.
<code>sort</code>	A list of metrics and dimension for sorting result data, for desc sorting use -.
<code>date.from</code>	Start date in YYYY-MM-DD.
<code>date.to</code>	End date in YYYY-MM-DD.
<code>accuracy</code>	Sampling level, one of low, medium, high or full. Or numeric from 0 to 1, where 1 - no sampling.
<code>include.undefined</code>	Includes in response rows for which grouping values are not defined.
<code>lang</code>	Language, by default ru
<code>timezone</code>	Timezone for time data.
<code>pretty</code>	Specifies the formatting of the result.
<code>login</code>	character, Your Yandex login
<code>token.path</code>	character, Directory for store API credential

Details

This API have some limits. 1. 10 demensions in the query. 2. 20 metrics in one query. 3. Limit: the number of unique groupings and metrics is up to 10, the number of individual filters is up to 20, the length of the line in the filter is up to 10,000 characters.

Value

Data frame with dimension and metrics.

Author(s)

Alexey Seleznev

References

[Reporting API docs](#)

Run `vignette("rym-reporting-api", package = "rym")` to see the corresponding vignette.

See Also

For load statics you also can use [rym_get_ga](#) and [rym_get_logs](#)

rym_get_filters *List of filters*

Description

Returns information about counter filters.

Usage

```
rym_get_filters(counter, login = getOption("rym.user"),
               token.path = getOption("rym.token_path"))
```

Arguments

counter	Yandex Metrika counter ID, you can get ids of all your counters over rym_counters
login	character, Your Yandex login
token.path	character, Directory for store API credential, by default is your work directory

Value

Data frame with list of yandex metrika counter filters

id	Filter id.
attr	The type of data to which the filter applies. One of referer, uniq_id, client_ip, title, url.
type	The relation or action for the filter, equal, contain, me, start, interval, only_mirrors.
value	Filter value.
action	Type of filter, include or exclude.
status	Status of filter, active or disabled.
with_subdomains	Filter by subdomains.
start_ip	The first IP address of the range.
end_ip	The last IP address of the range.

Author(s)

Alexey Seleznev

References

[Official docs](#)

Examples

```
## Not run:
filters <- rym_get_filters(counter = "your_counter_id",
                          login   = "your_login")

## End(Not run)
```

rym_get_ga	<i>Work with 'compatible with the Google Analytics Core Reporting API (v3)'</i>
------------	---

Description

Allows you to perform the following operations: 1. Obtain information about site traffic and other data. 2. Integrate Yandex.Metrica data with applications developed with the 'Google Analytics Core Reporting API (v3)' in mind. 3. Use the usual query parameters when collecting statistics, if you previously worked with the Google Analytics Core Reporting API (v3).

Usage

```
rym_get_ga(start.date = "10daysAgo", end.date = "today",
           counter = NULL, dimensions = "ga:date,ga:sourceMedium",
           metrics = "ga:sessions,ga:bounces,ga:users",
           filters = NULL, sort = NULL,
           sampling.level = "HIGHER_PRECISION",
           login = getOption("rym.user"),
           token.path = getOption("rym.token_path"))
```

Arguments

start.date	Start date in format YYYY-MM-DD
end.date	End date in format YYYY-MM-DD
counter	Your yandex metrica counter
dimensions	List of dimensions fields, for example "ga:browser,ga:city", see dictionary of available dimension
metrics	List of metrics fields, for example "ga:users,ga:sessions", see dictionary of available metrics
filters	List of filtrind dimensions and metrics.
sort	Sorting fields
sampling.level	One of "HIGHER_PRECISION", "FASTER", "DEFAULT"
login	character, Your Yandex login
token.path	character, Directory for store API credential

Details

Limits: 10 metrics for one query. 7 dimensions for one query

Value

Data frame with fields.

Author(s)

Alexey Seleznev

References

[Official compatible with the Google Analytics Core Reporting API \(v3\) docs](#)

Run vignette("rym-ga-api", package = "rym") to see the corresponding vignette.

See Also

For load statics you also can use [rym_get_data\(\)](#) and [rym_get_logs\(\)](#)

Examples

```
## Not run:
data <- rym_get_ga(start.date = "2017-08-01",
                  end.date = "yesterday",
                  counter = "ga:26841129",
                  metrics = "ga:sessions,ga:bounces,ga:users",
                  dimensions = "ga:date,ga:sourceMedium",
                  login      = "my_login")

## End(Not run)
```

rym_get_goals

List of goals

Description

Returns information about the purpose of the yandex metrika counter.

Usage

```
rym_get_goals(counter, login = getOption("rym.user"),
              token.path = getOption("rym.token_path"))
```

Arguments

counter	Yandex Metrika counter ID
login	character, Your Yandex login
token.path	character, Directory for store API credential

Value

Data frame with list of yandex metrika counters

Author(s)

Alexey Seleznev

References

[Official docs of 'Yandex Metrika Management API'](#)

[Docs on rym website](#)

Run `vignette("rym-management-api", package = "rym")` to see the corresponding vignette.

Examples

```
## Not run:
my_counters <- rym_get_goals(login = "my_login")

## End(Not run)
```

<code>rym_get_logs</code>	<i>Get raw data from yandex metrika.</i>
---------------------------	--

Description

'Logs API' allows you to receive non-aggregated data collected by Yandex.Metrica. This API is intended for service users who want to independently process statistical data and use them to solve unique analytical problems.

Usage

```
rym_get_logs(counter = NULL, date.from = Sys.Date() - 10, date.to =
  Sys.Date() - 1, fields =
  "ym:s:date,ym:s:counterID,ym:s:dateTime,
ym:s:isNewUser,ym:s:startURL,ym:s:visitDuration,
ym:s:ipAddress,ym:s:referrer",
  source = "visits", login = getOption("rym.user"),
  token.path = getOption("rym.token_path"))
```

Arguments

<code>counter</code>	Yandex metrika counter id
<code>date.from</code>	Start date in format YYYY-MM-DD
<code>date.to</code>	End date in format YYYY-MM-DD
<code>fields</code>	List of fields
<code>source</code>	Log source, one of "hits" or "visits"
<code>login</code>	character, Your Yandex login
<code>token.path</code>	character, Directory for store API credential

Value

Data frame with values and fields

Author(s)

Alexey Seleznev

References

[Visits fields](#)

[Hits fields](#)

[Logs API docs](#)

[Docs on rym website](#)

Run `vignette("rym-logs-api", package = "rym")` to see the corresponding vignette.

See Also

For load statics you also can use [rym_get_data\(\)](#) and [rym_get_ga\(\)](#)

Examples

```
## Not run:
# where "00000000" is your counter id
rawmetrikdata <- rym_get_logs(counter = "00000000",
                             date.from = "2016-12-01",
                             date.to = "2016-12-20",
                             fields = "ym:s:visitID,
                                     ym:s:date,
                                     ym:s:bounce,
                                     ym:s:clientID,
                                     ym:s:networkType",
                             source = "visits")

## End(Not run)
```

<code>rym_get_my_logins</code>	<i>Show list of auth logins</i>
--------------------------------	---------------------------------

Description

Get login list, chose and set default yandex login in current R session.

Usage

```
rym_get_my_logins(token.path = getOption("rym.token_path"),
                  set.login = TRUE)
```

Arguments

token.path character, Directory for store API credential
 set.login logical, if TRUE you set one of your account as session default

Author(s)

Alexey Seleznev

Examples

```
## Not run:
rym_get_my_logins()

## End(Not run)
```

rym_get_segments *List of segments*

Description

Returns a list of segments created using the API and related to the specified counter.

Usage

```
rym_get_segments(counter, login = getOption("rym.user"),
                 token.path = getOption("rym.token_path"))
```

Arguments

counter Yandex Metrika counter ID, you can get ids of all your counters over rym_counters
 login character, Your Yandex login
 token.path character, Directory for store API credential, by default is your work directory

Value

Data frame with list of yandex metrika counter segments

id Segment id.
 counter_id Yandex Metrika counter id.
 name Segment name.
 expression An expression that matches the value of the filters parameter.
 is_retargeting Logical, is retargeting segment or not.
 segment_source Source of the segment. Indicates how to create it. Accepts the value of api - segments that are created using the API are used.

Author(s)

Alexey Seleznev

References

[Official docs of Yandex Metrika Management API](#)

[Docs on rym website](#)

Run vignette("rym-management-api", package = "rym") to see the corresponding vignette.

Examples

```
## Not run:
segments <- rym_get_segments(counter = "your_counter_id",
                             login   = "your_login")

## End(Not run)
```

rym_users_grants	<i>List of users permissions</i>
------------------	----------------------------------

Description

Returns information about the permissions to manage the counter and viewing statistics from 'management API'.

Usage

```
rym_users_grants(counter, login = getOption("rym.user"),
                 token.path = getOption("rym.token_path"))
```

Arguments

counter	Yandex Metrika counter ID, you can get ids of all your counters over rym_counters
login	character, Your Yandex login
token.path	character, Directory for store API credential, by default is your work directory

Value

Data frame with list of yandex metrika counter users

user_login	Login of the user who has been given permission to manage the counter.
perm	Access level, one of view, edit, public_stat.
created_at	Date of granting access in the format YYYY-MM-DD'T'hh: mm: ssZ.
comment	An arbitrary comment. The number of characters can not exceed 255.
partner_data_access	Logical, is partner access or not.

Author(s)

Alexey Seleznev

References

[Official docs](#)

Run `vignette("rym-management-api", package = "rym")` to see the corresponding vignette.

Examples

```
## Not run:  
users <- rym_users_grants(counter = "your_counter_id",  
                          login   = "your_login")  
  
## End(Not run)
```

Index

rym (rym-package), 2
rym-calls, 4
rym-expense-uploading, 7
rym-offline-conversion, 9
rym-package, 2
rym_add_goal, 11
rym_add_goal(), 13
rym_add_segment, 12
rym_allow_calls (rym-calls), 4
rym_allow_offline_conversion
 (rym-offline-conversion), 9
rym_auth, 13
rym_delete_uploaded_expense
 (rym-expense-uploading), 7
rym_disable_calls (rym-calls), 4
rym_disable_offline_conversion
 (rym-offline-conversion), 9
rym_enable_calls (rym-calls), 4
rym_enable_offline_conversion
 (rym-offline-conversion), 9
rym_get_counters, 14
rym_get_data, 15
rym_get_data(), 20, 22
rym_get_direct_clients, 17
rym_get_filters, 18
rym_get_ga, 16, 19
rym_get_ga(), 22
rym_get_goals, 20
rym_get_logs, 16, 21
rym_get_logs(), 20
rym_get_my_logins, 22
rym_get_segments, 23
rym_get_uploadings_calls (rym-calls), 4
rym_get_uploadings_expense
 (rym-expense-uploading), 7
rym_get_uploadings_offline_conversions
 (rym-offline-conversion), 9
rym_upload_calls (rym-calls), 4
rym_upload_expense
 (rym-expense-uploading), 7
rym_upload_offline_conversion
 (rym-offline-conversion), 9
rym_users_grants, 24