

# Package ‘rym’

August 12, 2020

**Type** Package

**Title** R Interface to Yandex Metrica API

**Version** 1.0.3

**Date** 2020-08-12

**Author** Alexey Seleznev [aut, cre] (<<https://orcid.org/0000-0003-0410-7385>>)

**Maintainer** Alexey Seleznev <[selesnow@gmail.com](mailto: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** <https://selesnow.github.io/rym/>

**Encoding** UTF-8

**Language** ru

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-08-12 16:20:03 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 . . . . .	13
rym_auth . . . . .	14
rym_get_counters . . . . .	15
rym_get_data . . . . .	16
rym_get_direct_clients . . . . .	17
rym_get_filters . . . . .	18
rym_get_ga . . . . .	19
rym_get_goals . . . . .	21
rym_get_logs . . . . .	22
rym_get_my_logins . . . . .	23
rym_get_segments . . . . .	24
rym_users_grants . . . . .	25
<b>Index</b>	<b>26</b>

---

rym-package	<i>R Interface to Yandex Metrika API</i>
-------------	--

---

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:	1.0.3
Date:	2020-08-12
Authors@R:	c(person(given = "Alexey", family = "Seleznev", role = c("aut", "cre"), email = "selesnow@gmail.com", co
Author:	Alexey Seleznev [aut, cre] (< <a href="https://orcid.org/0000-0003-0410-7385">https://orcid.org/0000-0003-0410-7385</a> >)
Maintainer:	Alexey Seleznev < <a href="mailto:selesnow@gmail.com">selesnow@gmail.com</a> >
Description:	Allows work with 'Management API' for load counters, segments, filters, user permissions and goals list fr

Depends: R ( $\geq$  3.5.0)  
 BugReports: <https://github.com/selesnow/rym/issues>  
 Suggests: knitr, rmarkdown  
 VignetteBuilder: knitr  
 License: GPL-2  
 Imports: httr, stringr, utils, purrr  
 URL: <https://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 [aut, cre] (<<https://orcid.org/0000-0003-0410-7385>>)

Maintainer: Alexey Seleznev <[selesnow@gmail.com](mailto: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)
```

## 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

<code>new.goal.name</code>	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
<code>login</code>	character, Your Yandex login
<code>token.path</code>	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 Metrika Calls in Yandex Metrika](#)

## 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 exmaple cpc

**UTMCampaign** Campaign name, for example my campaing

**UTMTerm** Term name, for example my term

**UTMContent** Content name, for example my contant

**Currency** Currency in [ISO 4217](#) code

**Clicks** Clicks number, for example 23

**Author(s)**

Alexey Seleznev

**References**

See official dox: [API documents](#) [Yandex Metrica 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 doc: [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 <a href="#">link</a>
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

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

**Examples**

```
## Not run:
test_data <- rym_get_data(counters = 26841129,
                          dimensions = "ym:s:date,
                                       ym:s:lastsignTrafficSource",
                          login = "my_login",
                          sort = "ym:s:date")

## End(Not run)
```

---

`rym_get_direct_clients`*Load Yandex.Direct clients*

---

**Description**

Returns data about Yandex.Direct clients for campaigns that the owner of the Metrics counter has access to.

**Usage**

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

**Arguments**

counters	Counter identifiers, separated by commas
login	character, Your Yandex login
token.path	character, Directory for store API credential

**Value**

Data frame with Yandex Direct clients.

**Author(s)**

Alexey Seleznev

**References**

[Reporting API docs](#)

**Examples**

```
## Not run:
my_clients <- rym_get_direct_clients(counters="11111,22222",
                                     login = "my_login")

## End(Not run)
```

---

rym_get_filters	<i>List of filters</i>
-----------------	------------------------

---

**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.
start_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 metrika counter
dimensions	List of dimensions fields, for example "ga:browser,ga:city", see <a href="#">dictionary of available dimension</a>
metrics	List of metrics fields, for example "ga:users,ga:sessions", see <a href="#">dictionary of available metrics</a>
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	<i>List of goals</i>
---------------	----------------------

---

**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)
```

---

rym\_get\_logs

---

*Get raw data from yandex metrika.*


---

## 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

counter	Yandex metrika counter id
date.from	Start date in format YYYY-MM-DD
date.to	End date in format YYYY-MM-DD
fields	List of fields
source	Log source, one of "hits" or "visits"
login	character, Your Yandex login
token.path	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)
```

---

rym_get_my_logins	<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	<i>List of segments</i>
------------------	-------------------------

---

**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, 13  
rym\_allow\_calls (rym-calls), 4  
rym\_allow\_offline\_conversion  
    (rym-offline-conversion), 9  
rym\_auth, 14  
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, 15  
rym\_get\_data, 16  
rym\_get\_data(), 20, 23  
rym\_get\_direct\_clients, 17  
rym\_get\_filters, 18  
rym\_get\_ga, 17, 19  
rym\_get\_ga(), 23  
rym\_get\_goals, 21  
rym\_get\_logs, 17, 22  
rym\_get\_logs(), 20  
rym\_get\_my\_logins, 23  
rym\_get\_segments, 24  
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, 25