

Package ‘googleAnalyticsR’

February 16, 2018

Type Package

Version 0.5.0

Title Google Analytics API into R

Description Interact with the Google Analytics APIs <<https://developers.google.com/analytics/>>, including the Core Reporting API (v3 and v4), Management API, and Multi-Channel Funnel API.

URL <http://code.markedmondson.me/googleAnalyticsR/>

BugReports <https://github.com/MarkEdmondson1234/googleAnalyticsR/issues>

Depends R (>= 3.2.0)

Imports assertthat (>= 0.2.0), dplyr (>= 0.7.0), googleAuthR (>= 0.6.2), httr (>= 1.3.1), magrittr (>= 1.5), memoise, purrr (>= 0.2.2), rlang (>= 0.1.0), tidyr (>= 0.6.3), utils, methods

Suggests bigQueryR (>= 0.3.1), covr, googleCloudStorageR (>= 0.2.0), httpptest, knitr, miniUI (>= 0.1.1), rmarkdown, shiny (>= 0.13.2)

License MIT + file LICENSE

LazyData TRUE

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Mark Edmondson [aut, cre],
Artem Klevtsov [ctb],
Johann deBoer [ctb],
David Watkins [ctb],
Olivia Brode-Roger [ctb],
Jas Sohi [ctb],
Zoran Selinger [ctb]

Maintainer Mark Edmondson <m@sunhola.com>

Repository CRAN

Date/Publication 2018-02-16 11:19:00 UTC

R topics documented:

aggregateGAData	3
allowed_metric_dim	4
authDropdown	4
authDropdownUI	5
dim_filter	6
fetch_google_analytics_4	7
fetch_google_analytics_4_slow	8
filter_clause_ga4	9
ga_accounts	10
ga_adwords	11
ga_adwords_list	12
ga_auth	13
ga_cache_call	14
ga_custom_datasource	15
ga_custom_upload	15
ga_custom_upload_file	17
ga_custom_upload_list	18
ga_custom_vars	19
ga_custom_vars_list	20
ga_experiment	20
ga_experiment_list	21
ga_filter	22
ga_filter_add	22
ga_filter_apply_to_view	24
ga_filter_delete	25
ga_filter_list	26
ga_filter_update	27
ga_filter_update_filter_link	28
ga_filter_view	29
ga_filter_view_list	30
ga_goal	31
ga_goal_list	32
ga_remarketing_estimate	32
ga_remarketing_get	33
ga_remarketing_list	34
ga_segment_list	35
ga_unsampled	35
ga_unsampled_download	36
ga_unsampled_list	37
ga_users_list	38
ga_view	39
ga_view_list	40
ga_webproperty	40
ga_webproperty_list	41
getColNameOfClass	42
googleAnalyticsR	42

- google_analytics 43
- google_analytics_3 45
- google_analytics_account_list 48
- google_analytics_bq 48
- google_analytics_meta 50
- make_cohort_group 50
- make_ga_4_req 51
- meta 54
- met_filter 54
- multi_select 56
- multi_selectUI 57
- order_type 57
- pivot_ga4 58
- segmentBuilder 59
- segmentBuilderUI 60
- segment_define 61
- segment_element 62
- segment_ga4 63
- segment_vector_sequence 66
- segment_vector_simple 66

Index **68**

aggregateGAData *Aggregate a Google Analytics dataframe over inputted columns*

Description

A helper function to aggregate over dimensions

Usage

```
aggregateGAData(ga_data, agg_names = NULL,
  mean_regex = "^avg|^percent|Rate$|^CPC$|^CTR$|^CPM$|^RPC$|^ROI$|^ROAS$|Per")
```

Arguments

- ga_data A dataframe of data to aggregate
- agg_names The columns to aggregate over
- mean_regex The regex for column names to do mean() rather than sum()

Details

Will auto select metrics if they are numeric class columns. Will auto perform mean aggregation if metric names match mean_regex argument If agg_names is NULL will aggregate over all

allowed_metric_dim *Create named list of allowed GA metrics/dimensions*

Description

Create named list of allowed GA metrics/dimensions

Usage

```
allowed_metric_dim(type = c("METRIC", "DIMENSION"), subType = c("all",
  "segment", "cohort"), callAPI = FALSE)
```

Arguments

type	Type of parameter to create
subType	to restrict to only those in this type
callAPI	This will update the meta table (Requires online authorization) This is useful to expand goalXCompletions to all the possibilities, as well as restricting to those that variables that work with your API call. Use internal meta table, but you have option to update to the latest version.

Value

A named list of parameters for use in API calls

authDropdown *authDropdown [Shiny Module]*

Description

Shiny Module for use with [authDropdownUI](#)

Usage

```
authDropdown(input, output, session, ga.table)
```

Arguments

input	shiny input
output	shiny output
session	shiny session
ga.table	A table of GA tables

Details

Call via `shiny::callModule(authDropdown, "your_id")`

Value

GA View Id selected

See Also

Other Shiny modules: [authDropdownUI](#), [multi_selectUI](#), [multi_select](#)

authDropdownUI

authDropdown UI [Shiny Module]

Description

Makes a dropdown row for use for authentication.

Usage

```
authDropdownUI(id, width = NULL, inColumns = FALSE)
```

Arguments

<code>id</code>	Shiny id.
<code>width</code>	The width of the input
<code>inColumns</code>	whether to wrap selectInputs in width=4 columns. Shiny Module for use with authDropdown .

Value

Shiny UI

See Also

Other Shiny modules: [authDropdown](#), [multi_selectUI](#), [multi_select](#)

dim_filter	<i>Make a dimension filter object</i>
------------	---------------------------------------

Description

Make a dimension filter object

Usage

```
dim_filter(dimension, operator = c("REGEXP", "BEGINS_WITH", "ENDS_WITH",  
  "PARTIAL", "EXACT", "NUMERIC_EQUAL", "NUMERIC_GREATER_THAN",  
  "NUMERIC_LESS_THAN", "IN_LIST"), expressions, caseSensitive = FALSE,  
  not = FALSE)
```

Arguments

dimension	dimension name to filter on.
operator	How to match the dimension.
expressions	What to match. A character vector if operator is "IN_LIST"
caseSensitive	Boolean.
not	Logical NOT operator. Boolean.

Value

An object of class `dim_fil_ga4` for use in [filter_clause_ga4](#)

See Also

Other filter functions: [filter_clause_ga4](#), [met_filter](#)

Examples

```
## Not run:  
library(googleAnalyticsR)  
  
## authenticate,  
## or use the RStudio Addin "Google API Auth" with analytics scopes set  
ga_auth()  
  
## get your accounts  
account_list <- google_analytics_account_list()  
  
## pick a profile with data to query  
  
ga_id <- account_list[23,'viewId']  
  
## create filters on metrics
```

```
mf <- met_filter("bounces", "GREATER_THAN", 0)
mf2 <- met_filter("sessions", "GREATER", 2)

## create filters on dimensions
df <- dim_filter("source", "BEGINS_WITH", "1", not = TRUE)
df2 <- dim_filter("source", "BEGINS_WITH", "a", not = TRUE)

## construct filter objects
fc2 <- filter_clause_ga4(list(df, df2), operator = "AND")
fc <- filter_clause_ga4(list(mf, mf2), operator = "AND")

## make v4 request
ga_data1 <- google_analytics_4(ga_id,
                              date_range = c("2015-07-30", "2015-10-01"),
                              dimensions=c('source', 'medium'),
                              metrics = c('sessions', 'bounces'),
                              met_filters = fc,
                              dim_filters = fc2,
                              filtersExpression = "ga:source!=(direct)")

## End(Not run)
```

fetch_google_analytics_4

Fetch multiple GAv4 requests

Description

Fetch the GAv4 requests as created by [make_ga_4_req](#)

Usage

```
fetch_google_analytics_4(request_list, merge = FALSE,
                          useResourceQuotas = NULL)
```

Arguments

`request_list` A list of requests created by [make_ga_4_req](#)

`merge` If TRUE then will rbind that list of data.frames

`useResourceQuotas` If using GA360, access increased sampling limits. Default NULL, set to TRUE or FALSE if you have access to this feature.

Details

For same viewId, daterange, segments, samplingLevel and cohortGroup, v4 batches can be made

Value

A dataframe if one request, or a list of data.frames if multiple.

See Also

Other GAv4 fetch functions: [fetch_google_analytics_4_slow](#), [google_analytics](#), [make_ga_4_req](#)

Examples

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

## authenticate,
## or use the RStudio Addin "Google API Auth" with analytics scopes set
ga_auth()

## get your accounts
account_list <- google_analytics_account_list()

## pick a profile with data to query

ga_id <- account_list[23, 'viewId']

ga_req1 <- make_ga_4_req(ga_id,
                        date_range = c("2015-07-30", "2015-10-01"),
                        dimensions=c('source', 'medium'),
                        metrics = c('sessions'))

ga_req2 <- make_ga_4_req(ga_id,
                        date_range = c("2015-07-30", "2015-10-01"),
                        dimensions=c('source', 'medium'),
                        metrics = c('users'))

fetch_google_analytics_4(list(ga_req1, ga_req2))

## End(Not run)
```

fetch_google_analytics_4_slow

Fetch GAv4 requests one at a time

Description

Due to large complicated queries causing the v4 API to timeout, this option is added to fetch via the more traditional one report per request

Usage

```
fetch_google_analytics_4_slow(request_list, max_rows, allRows = FALSE,
  useResourceQuotas = NULL)
```

Arguments

`request_list` A list of requests created by [make_ga_4_req](#)

`max_rows` Number of rows requested (if not fetched)

`allRows` Whether to fetch all available rows

`useResourceQuotas` If using GA360, access increased sampling limits. Default NULL, set to TRUE or FALSE if you have access to this feature.

Value

A dataframe of all the requests

See Also

Other GA4 fetch functions: [fetch_google_analytics_4](#), [google_analytics](#), [make_ga_4_req](#)

`filter_clause_ga4` *Make a dimension or metric filter clause object*

Description

Make a dimension or metric filter clause object

Usage

```
filter_clause_ga4(filters, operator = c("OR", "AND"))
```

Arguments

`filters` a list of [dim_filter](#) or [met_filter](#). Only one type allowed.

`operator` combination of filter.

Details

If you have dimension and metric filters, make the clauses in two separate calls, then pass the objects to [make_ga_4_req](#)

Value

An object of class `dim_fil_ga4` or `met_fil_ga4` for use in [make_ga_4_req](#)

See Also

Other filter functions: [dim_filter](#), [met_filter](#)

Examples

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

## authenticate,
## or use the RStudio Addin "Google API Auth" with analytics scopes set
ga_auth()

## get your accounts
account_list <- google_analytics_account_list()

## pick a profile with data to query

ga_id <- account_list[23,'viewId']

## create filters on metrics
mf <- met_filter("bounces", "GREATER_THAN", 0)
mf2 <- met_filter("sessions", "GREATER", 2)

## create filters on dimensions
df <- dim_filter("source", "BEGINS_WITH", "1", not = TRUE)
df2 <- dim_filter("source", "BEGINS_WITH", "a", not = TRUE)

## construct filter objects
fc2 <- filter_clause_ga4(list(df, df2), operator = "AND")
fc <- filter_clause_ga4(list(mf, mf2), operator = "AND")

## make v4 request
ga_data1 <- google_analytics_4(ga_id,
                              date_range = c("2015-07-30", "2015-10-01"),
                              dimensions=c('source', 'medium'),
                              metrics = c('sessions', 'bounces'),
                              met_filters = fc,
                              dim_filters = fc2,
                              filtersExpression = "ga:source!=(direct)")

## End(Not run)
```

Description

Get account list

Usage

```
ga_accounts()
```

Value

a dataframe of accounts

See Also

Other managementAPI functions: [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

 ga_adwords

Get AdWords Link meta data

Description

Get AdWords Link meta data

Usage

```
ga_adwords(accountId, webPropertyId, webPropertyAdWordsLinkId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
webPropertyAdWordsLinkId	AdWords Link Id

Value

AdWords Meta data

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

 ga_adwords_list

List AdWords

Description

List AdWords

Usage

```
ga_adwords_list(accountId, webPropertyId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id

Value

AdWords Links

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

`ga_auth`*Authenticate with Google Analytics OAuth2*

Description

A wrapper for [gar_auth](#) and [gar_auth_service](#)

Usage

```
ga_auth(token = NULL, new_user = FALSE, no_auto = FALSE)
```

Arguments

<code>token</code>	An existing token or file location of a token to authenticate with
<code>new_user</code>	If TRUE, reauthenticate via Google login screen
<code>no_auto</code>	Skip auto authentication

Details

Run this function first time to authenticate with Google in your browser.

After initial authentication, a `.httr-oauth` will be saved to your working directory, where your authentication details are kept. Keep this file safe.

If you want to reauthenticate, delete this file from your directory or run `ga_auth(new_user = TRUE)`

Value

Invisibly, the token that has been saved to the session

Multiple accounts

You can authenticate with a new auth file for each account. Supply argument `token` with the name of the cache file you want to use e.g. `ga_auth(token = "one.httr-oauth")` for one account, `ga_auth(token = "another.httr-oauth")` for a different account.

Auto-authentication

You can choose to auto-authenticate by moving your `.httr-oauth` or by creating a Google OAuth service account JSON file.

Specify an environment variable in R via a `.Renviron` file or using [Sys.setenv](#) which point to the file location of your chosen authentication file. See [Startup](#)

Once you have set the environment variable `GA_AUTH_FILE` to a valid file location, the function will look there for authentication details upon loading the library meaning you will not need to call `ga_auth()` yourself as you would normally.

An example `.Renviron` file is below:

```
GA_AUTH_FILE = "/Users/bob/auth/googleAnalyticsR.httr-oauth"
```

GA_AUTH_FILE can be either a auth file for a token generated by [gar_auth](#) or service account JSON ending with file extension .json

Your own Google Project

By default the Google Project used is shared by all users, so you may find it runs out of API calls. To mitigate that, create your own Google Project and turn on the Analytics APIs.

You can then create your own client ID and client secret, to place in options or environment arguments (whichever is easiest)

The environment args are below. Similar to auto-authentication, you can place your entries in an .Renvirom file

```
GA_CLIENT_ID="XXXX" GA_CLIENT_SECRET="XXX" GA_WEB_CLIENT_ID="XXX" GA_WEB_CLIENT_SECRET="XXX"
```

Service accounts

If you use the service account JSON, you will need to add the service account email to your Google Analytics users to see data e.g. xxxx@yyyyyy.iam.gserviceaccount.com

ga_cache_call

Setup caching of API calls

Description

Lets you cache API calls to disk

Usage

```
ga_cache_call(cache_location)
```

Arguments

cache_location If RAM will save to memory, or specify a file folder location

Details

By default this is turned on upon package load to RAM. Should you want to cache calls to a folder then run this function to specify where.

ga_custom_datasource *List Custom Data Sources*

Description

Get a list of custom data sources you have configured in Google Analytics web UI.

Usage

```
ga_custom_datasource(accountId, webPropertyId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id

Details

You primarily need this to get the customDataSourceId for the uploads via [ga_custom_upload_file](#)

Value

Custom Data Source

See Also

Other custom datasource functions: [ga_custom_upload_file](#), [ga_custom_upload_list](#), [ga_custom_upload](#)

ga_custom_upload *Custom Data Source Upload Status*

Description

Get the status of a custom upload

Usage

```
ga_custom_upload(accountId, webPropertyId, customDataSourceId, uploadId,  
upload_object)
```

Arguments

accountId Account Id
 webPropertyId Web Property Id
 customDataSourceId
 Custom data source Id
 uploadId upload Id
 upload_object A custom upload Id object. Supply this or the other arguments.

Details

You can supply either `upload_object` generated via function or [ga_custom_upload_file](#), or make an

Value

An object of class `ga_custom_data_source_upload`

See Also

Other custom datasource functions: [ga_custom_datasource](#), [ga_custom_upload_file](#), [ga_custom_upload_list](#)

Examples

```
## Not run:

upload_me <- data.frame(
  medium = "shinyapps",
  source = "referral",
  adCost = 1,
  date = "20160801")

obj <- ga_custom_upload_file(47850439,
  "UA-4748043-2",
  "_jDsJHSFSU-uw038Bh8fUg",
  upload_me)

## obj will initially have status = PENDING
obj
==Google Analytics Custom Data Source Upload==
Custom Data Source ID: _jDsJHSFSU-uw038Bh8fUg
Account ID:          47850439
Web Property Id:     UA-4748043-2
Upload ID:           7yHLAkeLSiK1zveVTiWzWA
Status:              PENDING

## Send obj to ga_custom_upload() to check and renew status
obj <- ga_custom_upload(upload_object = obj)
obj

==Google Analytics Custom Data Source Upload==
```



```
Custom Data Source ID: _jDsJHSFSU-uw038Bh8fUg
Account ID: 47850439
Web Property Id: UA-4748043-2
Upload ID: 7yHLAkeLSiK1zveVTiWzWA
Status: COMPLETED
```

```
## End(Not run)
```

ga_custom_upload_file *Upload data to Google Analytics*

Description

Upload external data up to 1GB to Google Analytics via the management API.

Usage

```
ga_custom_upload_file(accountId, webPropertyId, customDataSourceId, upload)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
customDataSourceId	Custom data source Id
upload	An R data.frame or a file path location (character)

Details

You need to create a custom data source in the web UI first.

If you are uploading an R data frame, the function will prefix the column names with "ga:" for you if necessary.

After upload check the status by querying data sources using [ga_custom_upload](#) and examining the status field.

Currently only supports simple uploads (not resumable).

Value

An object of class `ga_custom_data_source_upload`

See Also

A guide for preparing the data is available: [from Google here](#).

The dev guide for this function: [Data Import Developer Guide](#)

Other custom datasource functions: [ga_custom_datasource](#), [ga_custom_upload_list](#), [ga_custom_upload](#)

Examples

```
## Not run:

upload_me <- data.frame(medium = "shinyapps",
  source = "referral",
  adCost = 1,
  date = "20160801")

obj <- ga_custom_upload_file(47850439,
  "UA-4748043-2",
  "_jDsJHSFSU-uw038Bh8fUg",
  upload_me)

## obj will initially have status = PENDING
obj
==Google Analytics Custom Data Source Upload==
Custom Data Source ID:  _jDsJHSFSU-uw038Bh8fUg
Account ID:             47850439
Web Property Id:       UA-4748043-2
Upload ID:             7yHLAkeLSiK1zveVTiWzWA
Status:               PENDING

## Send obj to ga_custom_upload() to check and renew status
obj <- ga_custom_upload(upload_object = obj)
obj

==Google Analytics Custom Data Source Upload==
Custom Data Source ID:  _jDsJHSFSU-uw038Bh8fUg
Account ID:             47850439
Web Property Id:       UA-4748043-2
Upload ID:             7yHLAkeLSiK1zveVTiWzWA
Status:               COMPLETED

## End(Not run)
```

ga_custom_upload_list *List Custom Data Source Uploads*

Description

List Custom Data Source Uploads

Usage

```
ga_custom_upload_list(accountId, webPropertyId, customDataSourceId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
customDataSourceId	Custom data source Id

Value

Custom Data Source Uploads List

See Also

Other custom datasource functions: [ga_custom_datasource](#), [ga_custom_upload_file](#), [ga_custom_upload](#)

ga_custom_vars	<i>Get Custom Dimensions or Metrics</i>
----------------	---

Description

Get Custom Dimensions or Metrics

Usage

```
ga_custom_vars(accountId, webPropertyId, type = c("customMetrics",
"customDimensions"), customId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
type	A customMetric or customDimension
customId	The customMetricId or customDimensionId

Value

Custom Metric or Dimension meta data

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_custom_vars_list *List Custom Dimensions or Metrics*

Description

List Custom Dimensions or Metrics

Usage

```
ga_custom_vars_list(accountId, webPropertyId, type = c("customMetrics",
  "customDimensions"))
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
type	A customMetric or customDimension

Value

Custom Metric or Dimension List

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_experiment *Experiments Meta data*

Description

Experiments Meta data

Usage

```
ga_experiment(accountId, webPropertyId, profileId, experimentId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
profileId	Profile Id
experimentId	Experiment Id

Value

Experiment Meta Data

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_experiment_list *List Experiments*

Description

List Experiments

Usage

ga_experiment_list(accountId, webPropertyId, profileId)

Arguments

accountId	Account Id
webPropertyId	Web Property Id
profileId	Profile Id

Value

Experiments List

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_filter	<i>Get specific filter for account</i>
-----------	--

Description

Get specific filter for account

Usage

```
ga_filter(accountId, filterId)
```

Arguments

accountId	Account Id
filterId	Filter Id

Value

filter list

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_filter_add	<i>Create a new filter and add it to the view (optional).</i>
---------------	---

Description

Take a filter object and add and/or apply it so its live.

Usage

```
ga_filter_add(Filter, accountId, webPropertyId = NULL, viewId = NULL,  
linkFilter = FALSE)
```

Arguments

Filter	The Filter object to be added to the account or view. See examples.
accountId	Account Id of the account to add the Filter to
webPropertyId	Property Id of the property to add the Filter to
viewId	View Id of the view to add the Filter to
linkFilter	If TRUE will apply the Filter to the view. Needs propertyId and viewId to be set.

Details

If you don't set linkFilter=TRUE then the filter will only be created but not applied. You will find it listed in the admin panel Account > All Filters. You can then use [ga_filter_apply_to_view](#) to apply later on.

Value

The filterId created if linkFilter=FALSE or a Filter object if linkFilter=TRUE

See Also

<https://developers.google.com/analytics/devguides/config/mgmt/v3/mgmtReference/#Filters>

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

Examples

```
## Not run:
## Create a filter object for adding an IP exclusion:
Filter <- list(
  name = 'Exclude Internal Traffic',
  type = 'EXCLUDE',
  excludeDetails = list(
    field = 'GEO_IP_ADDRESS',
    matchType = 'EQUAL',
    expressionValue = '199.04.123.1',
    caseSensitive = 'False'
  )
)

# create and add the filter to the view specified
my_filter <- ga_filter_add(Filter,
  accountId = 12345,
  webPropertyId = "UA-12345-1",
  viewId = 654321,
  linkFilter = TRUE)
```

```
# only create the filter, don't apply it to any view - returns filterId for use later
my_filter <- ga_filter_add(Filter,
                           accountId = 12345,
                           linkFilter = FALSE)

## Other examples of filters you can create below:
## Create a filter object for making campaign medium lowercase
Filter <- list(
  name = 'Lowercase Campaign Medium',
  type = 'LOWERCASE',
  lowercaseDetails = list(
    field = 'CAMPAIGN_MEDIUM'
  )
)

## Create a filter object to append hostname to URI
Filter <- list(
  name = 'Append hostname to URI',
  type = 'ADVANCED',
  advancedDetails = list(
    fieldA = 'PAGE_HOSTNAME',
    extractA = '(.*)',
    fieldARequired = 'True',
    fieldB = 'PAGE_REQUEST_URI',
    extractB = '(.*)',
    fieldBRequired = 'False',
    outputConstructor = '$A1$B1',
    outputToField = 'PAGE_REQUEST_URI',
    caseSensitive = 'False',
    overrideOutputField = 'True'
  )
)

## Create a filter object to add www hostname without it
Filter <- list(
  name = 'Search and Replace www',
  type = 'SEARCH_AND_REPLACE',
  searchAndReplaceDetails = list(
    field = 'PAGE_HOSTNAME',
    searchString = '^exampleUSA\\.com$',
    replaceString = 'www.exampleUSA.com',
    caseSensitive = 'False'
  )
)

## End(Not run)
```

ga_filter_apply_to_view

Apply an existing filter to view.

Description

Apply an existing filter to view.

Usage

```
ga_filter_apply_to_view(filterId, accountId, webPropertyId, viewId)
```

Arguments

filterId	The id of the filter to be addedd to profile/view
accountId	Account Id of the account that contains the filter
webPropertyId	Web property Id to create profile filter link for
viewId	Profile/view Id to create profile filter link for

Value

A profileFilterLink object

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_filter_delete

Delete a filter from account or remove from view.

Description

Delete a filter from account or remove from view.

Usage

```
ga_filter_delete(accountId, webPropertyId = NULL, viewId = NULL, filterId,  
removeFromView = FALSE)
```

Arguments

accountId	Account Id of the account that contains the filter
webPropertyId	Property Id of the property that contains the filter
viewId	View Id of the view that contains the filter
filterId	Filter Id of the filter to be deleted
removeFromView	Default if FALSE. If TRUE, deletes the filter from the view

Value

TRUE if successful

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_filter_list	<i>List filters for account</i>
----------------	---------------------------------

Description

List filters for account

Usage

```
ga_filter_list(accountId)
```

Arguments

accountId	Account Id
-----------	------------

Value

filter list

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_filter_update	<i>Updates an existing filter.</i>
------------------	------------------------------------

Description

Updates an existing filter.

Usage

```
ga_filter_update(Filter, accountId, filterId, method = c("PUT", "PATCH"))
```

Arguments

Filter	The Filter object to be updated See examples from ga_filter_add()
accountId	Account Id of the account that contains the filter
filterId	The id of the filter to be modified
method	PUT by default. For patch semantics use PATCH

Value

A filterManagement object

See Also

<https://developers.google.com/analytics/devguides/config/mgmt/v3/mgmtReference/#Filters>

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

Examples

```
## Not run:

# create a filter object
Filter <- list(
  name = 'googleAnalyticsR test1: Exclude Internal Traffic',
  type = 'EXCLUDE',
  excludeDetails = list(
    field = 'GEO_IP_ADDRESS',
    matchType = 'EQUAL',
    expressionValue = '199.04.123.1',
    caseSensitive = 'False'
  )
)
```

```

    )
# add a filter (but don't link to a View)
filterId <- ga_filter_add(Filter,
                          accountId = 123456,
                          linkFilter = FALSE)

# change the name of the filter
change_name <- "googleAnalyticsR test2: Changed name via PATCH"

# using PATCH semantics, only need to construct what you want to change
filter_to_update <- list(name = test_name)

# update the filter using the filterId
ga_filter_update(filter_to_update, accountId2, filterId, method = "PATCH")

## End(Not run)

```

ga_filter_update_filter_link

Update an existing profile filter link. Patch semantics supported

Description

Update an existing profile filter link. Patch semantics supported

Usage

```
ga_filter_update_filter_link(viewFilterLink, accountId, webPropertyId, viewId,
                             linkId, method = c("PUT", "PATCH"))
```

Arguments

viewFilterLink	The profileFilterLink object
accountId	Account Id of the account that contains the filter
webPropertyId	Web property Id to which the profile filter link belongs
viewId	View Id to which the profile filter link belongs
linkId	The id of the profile filter link to be updated
method	PUT by default. Supports patch semantics when set to PATCH

See Also

<https://developers.google.com/analytics/devguides/config/mgmt/v3/mgmtReference/management/profileFilterLinks>

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#),

```
ga_filter_delete, ga_filter_list, ga_filter_update, ga_filter_view_list, ga_filter_view,  
ga_filter, ga_goal_list, ga_goal, ga_remarketing_estimate, ga_remarketing_get, ga_remarketing_list,  
ga_segment_list, ga_unsampled_list, ga_unsampled, ga_users_list, ga_view_list, ga_view,  
ga_webproperty_list, ga_webproperty, google_analytics_account_list
```

Examples

```
## Not run:  
  
# create a filter object  
Filter <- list(  
  name = 'googleAnalyticsR test: Exclude Internal Traffic',  
  type = 'EXCLUDE',  
  excludeDetails = list(  
    field = 'GEO_IP_ADDRESS',  
    matchType = 'EQUAL',  
    expressionValue = '199.04.123.1',  
    caseSensitive = 'False'  
  )  
)  
  
# link Filter to a View  
response <- ga_filter_add(Filter,  
  accountId = 12345,  
  webPropertyId = "UA-12345-1",  
  viewId = 654321,  
  linkFilter = TRUE)  
  
# create Filter patch to move existing filter up to rank 1  
viewFilterLink <- list(rank = 1)  
  
# use the linkId given in response$id to update to new rank 1  
response2 <- ga_filter_update_filter_link(viewFilterLink,  
  accountId = 12345,  
  webPropertyId = "UA-12345-1",  
  viewId = 654321,  
  linkId = response$id)  
  
## End(Not run)
```

ga_filter_view

Get specific filter for view (profile)

Description

Get specific filter for view (profile)

Usage

```
ga_filter_view(accountId, webPropertyId, viewId, linkId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
viewId	Profile Id
linkId	Link Id

Value

filter list

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_filter_view_list *List filters for view (profile)*

Description

List filters for view (profile)

Usage

```
ga_filter_view_list(accountId, webPropertyId, viewId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
viewId	Profile Id

Value

filter list

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

 ga_goal

Get goal

Description

Get goal

Usage

```
ga_goal(accountId, webPropertyId, profileId, goalId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
profileId	Profile Id
goalId	Goal Id

Value

Goal meta data

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_goal_list	<i>List goals</i>
--------------	-------------------

Description

List goals

Usage

```
ga_goal_list(accountId, webPropertyId, profileId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
profileId	Profile Id

Value

Goal list

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_remarketing_estimate	<i>Estimate number of users added to the segment yesterday</i>
-------------------------	--

Description

Estimate number of users added to the segment yesterday

Usage

```
ga_remarketing_estimate(remarketingAudience)
```


Arguments

remarketingAudience

A remarketing audience object from [ga_remarketing_get](#)

Takes the segment definition from a remarketing audiences and runs it against the viewId to see current estimated users

The total audience size is this figure for every membershipDurationDay from yesterday

Value

data.frame

See Also**About remarketing audiences**

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_remarketing_get *Get a remarketing audience*

Description

Get a remarketing audience

Usage

```
ga_remarketing_get(accountId, webPropertyId, remarketingAudienceId)
```

Arguments

accountId Account Id

webPropertyId Web Property Id

remarketingAudienceId

The ID of the remarketing audience to retrieve.

Value

Remarketing Audience object

See Also[About remarketing audiences](#)

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_remarketing_list	<i>List remarketing audiences</i>
---------------------	-----------------------------------

Description

List remarketing audiences

Usage

```
ga_remarketing_list(accountId, webPropertyId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id

Value

Remarketing audience list

See Also[About remarketing audiences](#)

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_segment_list	<i>Get segments user has access to</i>
-----------------	--

Description

Get segments user has access to

Usage

```
ga_segment_list()
```

Value

Segment list

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_unsampled	<i>Get Unsampled Report Meta Data</i>
--------------	---------------------------------------

Description

Get Unsampled Report Meta Data

Usage

```
ga_unsampled(accountId, webPropertyId, profileId, unsampledReportId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
profileId	Profile Id
unsampledReportId	Unsampled Report Id

Value

Unsampled Report Meta Data

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_unsampled_download *Download Unsampled Report from Google Drive*

Description

Download Unsampled Report from Google Drive

Usage

```
ga_unsampled_download(reportTitle, accountId, webPropertyId, profileId,
  downloadFile = TRUE)
```

Arguments

reportTitle	Title of Unsampled Report (case-sensitive)
accountId	Account Id
webPropertyId	Web Property Id
profileId	Profile Id
downloadFile	Default TRUE, whether to download, if FALSE returns a dataframe instead

Value

file location if downloadFile is TRUE, else a data.frame of download

Examples

```
## Not run:

# get data.frame of unsampled reports you have available
unsample_list <- ga_unsampled_list(accountId = "12345",
  webPropertyId = "UA-12345-4",
  profileId = "129371234")

# loop through unsampled reports and download as a list of data.frames
dl <- lapply(unsample_list$title, ga_unsampled_download,
  accountId = "12345",
  webPropertyId = "UA-12345-4",
  profileId = "129371234",
```

```
downloadFile = FALSE)

# inspect first data.frame
dl[[1]]

# download unsampled report to csv file
ga_unsampled_download("my_report_title",
                      accountId = "12345",
                      webPropertyId = "UA-12345-4",
                      profileId = "129371234")

## End(Not run)
```

ga_unsampled_list *List Unsampled Reports*

Description

List Unsampled Reports

Usage

```
ga_unsampled_list(accountId, webPropertyId, profileId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
profileId	Profile Id

Value

Unsampled Reports List

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

Examples

```
## Not run:

# get data.frame of unsampled reports you have available
unsample_list <- ga_unsampled_list(accountId = "12345",
                                   webPropertyId = "UA-12345-4",
                                   profileId = "129371234")

# loop through unsampled reports and download as a list of data.frames
dl <- lapply(unsample_list$title, ga_unsampled_download,
             accountId = "12345",
             webPropertyId = "UA-12345-4",
             profileId = "129371234",
             downloadFile = FALSE)

# inspect first data.frame
dl[[1]]

# download unsampled report to csv file
ga_unsampled_download("my_report_title",
                      accountId = "12345",
                      webPropertyId = "UA-12345-4",
                      profileId = "129371234")

## End(Not run)
```

ga_users_list

List Users

Description

Will list users on an account, webproperty or view level

Usage

```
ga_users_list(accountId, webPropertyId = NULL, viewId = NULL)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id - leave NULL to fetch account level only
viewId	viewId - leave NULL to fetch webProperty level only

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_view	<i>Get single View (Profile)</i>
---------	----------------------------------

Description

Get single View (Profile)

Usage

```
ga_view(accountId, webPropertyId, profileId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id
profileId	Profile (View) Id

Value

webproperty

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_view_list *List View (Profile)*

Description

List View (Profile)

Usage

```
ga_view_list(accountId, webPropertyId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id

Value

webproperty

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#), [google_analytics_account_list](#)

ga_webproperty *Get web property*

Description

Get web property

Usage

```
ga_webproperty(accountId, webPropertyId)
```

Arguments

accountId	Account Id
webPropertyId	Web Property Id

Value

webproperty

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [google_analytics_account_list](#)

ga_webproperty_list *List web properties*

Description

List web properties

Usage

ga_webproperty_list(accountId)

Arguments

accountId Account Id

Value

webproperty list

See Also

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty](#), [google_analytics_account_list](#)

`getColNameOfClass` *Gets the names of a dataframe's columns of a certain class*

Description

Gets the names of a dataframe's columns of a certain class

Usage

```
getColNameOfClass(df, class_name)
```

Arguments

<code>df</code>	dataframe
<code>class_name</code>	the R class to return columns of

Value

names of columns of class

`googleAnalyticsR` *Library for getting Google Analytics data into R*

Description

Follow the online documentation here: <http://code.markedmondson.me/googleAnalyticsR/>

Details

- First Google Analytics Reporting v4 API library for R
- v4 features include: dynamic calculated metrics, pivots, histograms, date comparisons, batching
- Multi-user login in Shiny App
- Integration with BigQuery Google Analytics Premium/360 exports
- Single authentication flow with other 'googleAuthR' apps
- Automatic batching
- sampling avoidance with daily walk
- multi-account fetching
- multi-channel funnels
- Support for batch. For big data calls this could be 10x quicker than normal GA fetching
- Meta data included in attributes of returned dataframe

You may wish to set the below environment arguments for easier authentication

```
GA_CLIENT_ID GA_CLIENT_SECRET GA_WEB_CLIENT_ID GA_WEB_CLIENT_SECRET GA_AUTH_FILE
```

google_analytics *Get Google Analytics v4 data*

Description

Fetch Google Analytics data using the v4 API. For the v3 API use [google_analytics_3](#).
Will perform automatic batching if over the 100000 row per API call limit.

Usage

```
google_analytics(viewId, date_range = NULL, metrics = NULL,
  dimensions = NULL, dim_filters = NULL, met_filters = NULL,
  filtersExpression = NULL, order = NULL, segments = NULL,
  pivots = NULL, cohorts = NULL, max = 1000,
  samplingLevel = c("DEFAULT", "SMALL", "LARGE"), metricFormat = NULL,
  histogramBuckets = NULL, anti_sample = FALSE,
  anti_sample_batches = "auto", slow_fetch = FALSE,
  useResourceQuotas = NULL, rows_per_call = 10000L)
```

```
google_analytics_4(...)
```

Arguments

viewId	viewId of data to get.
date_range	character or date vector of format c(start, end) or for two date ranges: c(start1,end1,start2,end2)
metrics	Metric to fetch. Supports calculated metrics.
dimensions	Dimensions to fetch.
dim_filters	A filter_clause_ga4 wrapping dim_filter
met_filters	A filter_clause_ga4 wrapping met_filter
filtersExpression	A v3 API style simple filter string. Not used with other filters.
order	An order_type object
segments	List of segments as created by segment_ga4
pivots	Pivots of the data as created by pivot_ga4
cohorts	Cohorts created by make_cohort_group
max	Maximum number of rows to fetch. Defaults at 1000. Use -1 to fetch all results. Ignored when anti_sample=TRUE.
samplingLevel	Sample level
metricFormat	If supplying calculated metrics, specify the metric type
histogramBuckets	For numeric dimensions such as hour, a list of buckets of data. See details in make_ga_4_req

<code>anti_sample</code>	If TRUE will split up the call to avoid sampling.
<code>anti_sample_batches</code>	"auto" default, or set to number of days per batch. 1 = daily.
<code>slow_fetch</code>	For large, complicated API requests this bypasses some API hacks that may result in 500 errors. For smaller queries, leave this as FALSE for quicker data fetching.
<code>useResourceQuotas</code>	If using GA360, access increased sampling limits. Default NULL, set to TRUE or FALSE if you have access to this feature.
<code>rows_per_call</code>	Set how many rows are requested by the API per call, up to a maximum of 100000.
<code>...</code>	Arguments passed to google_analytics

Value

A Google Analytics data.frame, with attributes showing row totals, sampling etc.

Row requests

By default the API call will use v4 batching that splits requests into 5 separate calls of 10k rows each. This can go up to 100k, so this means up to 500k rows can be fetched per API call, however the API servers will fail with a 500 error if the query is too complicated as the processing time at Google's end gets too long. In this case, you may want to tweak the `rows_per_call` argument downwards, or fall back to using `slow_fetch = FALSE` which will send an API request one at a time. If fetching data via scheduled scripts this is recommended as the default.

Anti-sampling

`anti_sample` being TRUE ignores `max` as the API call is split over days to mitigate the sampling session limit, in which case a row limit won't work. Take the top rows of the result yourself instead e.g. `head(ga_data_unsampled, 50300)`

`anti_sample` being TRUE will also set `samplingLevel='LARGE'` to minimise the number of calls.

Resource Quotas

If you are on GA360 and have access to resource quotas, set the `useResourceQuotas=TRUE` and set the Google Cloud client ID to the project that has resource quotas activated, via [gar_set_client](#) or options.

Caching

By default local caching is turned on for v4 API requests. This means that making the same request as one this session will read from memory and not make an API call. You can also set the cache to disk via the [gar_cache_setup](#) function. This can be useful when running RMarkdown reports using data. To empty the cache use [gar_cache_empty](#).

See Also

Other GAv4 fetch functions: [fetch_google_analytics_4_slow](#), [fetch_google_analytics_4](#), [make_ga_4_req](#)

Examples

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

## authenticate, or use the RStudio Addin "Google API Auth" with analytics scopes set
ga_auth()

## get your accounts

account_list <- google_analytics_account_list()

## account_list will have a column called "viewId"
account_list$viewId

## View account_list and pick the viewId you want to extract data from
ga_id <- 123456

## simple query to test connection
google_analytics(ga_id,
                 date_range = c("2017-01-01", "2017-03-01"),
                 metrics = "sessions",
                 dimensions = "date")

## End(Not run)
```

google_analytics_3 *Get Google Analytics v3 data (formerly google_analytics())*

Description

Legacy v3 API, for more modern API use [google_analytics](#).

Usage

```
google_analytics_3(id, start, end, metrics = c("sessions", "bounceRate"),
                  dimensions = NULL, sort = NULL, filters = NULL, segment = NULL,
                  samplingLevel = c("DEFAULT", "FASTER", "HIGHER_PRECISION", "WALK"),
                  max_results = 100, multi_account_batching = FALSE, type = c("ga",
                  "mcf"))
```

Arguments

id	A character vector of View Ids to fetch from.
start	Start date in YYYY-MM-DD format.
end	End date in YYYY-MM-DD format.
metrics	A character vector of metrics. With or without ga: prefix.
dimensions	A character vector of dimensions. With or without ga: prefix.
sort	How to sort the results, in form 'ga:sessions,-ga:bounceRate'
filters	Filters for the result, in form 'ga:sessions>0;ga:pagePath=~blah'
segment	How to segment.
samplingLevel	Choose "WALK" to mitigate against sampling.
max_results	Default 100. If greater than 10,000 then will batch GA calls.
multi_account_batching	If TRUE then multiple id's are fetched together. Not compatible with samplingLevel="WALK" or max_results>10000
type	ga = Google Analytics v3; mcf = Multi-Channel Funels.

Value

For one id a data.frame of data, with meta-data in attributes. For multiple id's, a list of dataframes.

See Also

<https://developers.google.com/analytics/devguides/reporting/core/v3/>

Examples

```
## Not run:

library(googleAnalyticsR)

## Authenticate in Google OAuth2
## this also sets options
ga_auth()

## if you need to re-authenticate use ga_auth(new_user=TRUE)
## if you have your own Google Dev console project keys,
## then don't run ga_auth() as that will set to the defaults.
## instead put your options here, and run googleAuthR::gar_auth()

## get account info, including View Ids
account_list <- google_analytics_account_list()
ga_id <- account_list$viewId[1]

## get a list of what metrics and dimensions you can use

meta <- google_analytics_meta()
```

```
head(meta)

## pick the account_list$viewId you want to see data for.
## metrics and dimensions can have or have not "ga:" prefix

gadata <- google_analytics(id = ga_id,
                           start="2015-08-01", end="2015-08-02",
                           metrics = c("sessions", "bounceRate"),
                           dimensions = c("source", "medium"))

## multi accounts, pass character vector of viewIds
## outputs a list of data.frames, named after the viewId
multi_gadata <- google_analytics(id = c("123456", "9876545", "765432"),
                                 start="2015-08-01", end="2015-08-02",
                                 metrics = c("sessions", "bounceRate"),
                                 dimensions = c("source", "medium"))

## if more than 10000 rows in results, auto batching
## example is setting lots of dimensions to try and create big sampled data
batch_gadata <- google_analytics(id = ga_id,
                                 start="2014-08-01", end="2015-08-02",
                                 metrics = c("sessions", "bounceRate"),
                                 dimensions = c("source", "medium",
                                                "landingPagePath",
                                                "hour", "minute"),
                                 max=99999999)

## mitigate sampling by setting samplingLevel="WALK"
## this will send lots and lots of calls to the Google API limits, beware
walk_gadata <- google_analytics(id = ga_id,
                                start="2014-08-01", end="2015-08-02",
                                metrics = c("sessions", "bounceRate"),
                                dimensions = c("source", "medium", "landingPagePath"),
                                max=99999999, samplingLevel="WALK")

## multi-channel funnels set type="mcf"
mcf_gadata <- google_analytics(id = ga_id,
                               start="2015-08-01", end="2015-08-02",
                               metrics = c("totalConversions"),
                               dimensions = c("sourcePath"),
                               type="mcf")

## reach meta-data via attr()
attr(gadata, "profileInfo")
attr(gadata, "dateRange")

## End(Not run)
```

`google_analytics_account_list`*Get account summary including the ViewId*

Description

Get account summary including the ViewId

Usage

```
google_analytics_account_list()
```

```
ga_account_list()
```

Value

a dataframe of all account, webproperty and view data

See Also

<https://developers.google.com/analytics/devguides/config/mgmt/v3/mgmtReference/management/accountSummaries/list>

Other managementAPI functions: [ga_accounts](#), [ga_adwords_list](#), [ga_adwords](#), [ga_custom_vars_list](#), [ga_custom_vars](#), [ga_experiment_list](#), [ga_experiment](#), [ga_filter_add](#), [ga_filter_apply_to_view](#), [ga_filter_delete](#), [ga_filter_list](#), [ga_filter_update_filter_link](#), [ga_filter_update](#), [ga_filter_view_list](#), [ga_filter_view](#), [ga_filter](#), [ga_goal_list](#), [ga_goal](#), [ga_remarketing_estimate](#), [ga_remarketing_get](#), [ga_remarketing_list](#), [ga_segment_list](#), [ga_unsampled_list](#), [ga_unsampled](#), [ga_users_list](#), [ga_view_list](#), [ga_view](#), [ga_webproperty_list](#), [ga_webproperty](#)

`google_analytics_bq` *Get Google Analytics 360 BigQuery data*

Description

Turn a google_analytics style call into BigQuery SQL. Used with Google Analytics 360 BigQuery exports.

Usage

```
google_analytics_bq(projectId, datasetId, start = NULL, end = NULL,  
  metrics = NULL, dimensions = NULL, sort = NULL, filters = NULL,  
  max_results = 100, query = NULL, return_query_only = FALSE,  
  bucket = NULL, download_file = NULL)
```


Arguments

projectId	The Google project Id where the BigQuery exports sit
datasetId	DatasetId of GA export. This should match the GA View ID
start	start date
end	end date
metrics	metrics to query
dimensions	dimensions to query
sort	metric to sort by
filters	filter results
max_results	How many results to fetch
query	If query is non-NULL then it will use that and ignore above
return_query_only	Only return the constructed query, don't call BigQuery
bucket	if over 100000 results, specify a Google Cloud bucket to send data to
download_file	Where to save asynch files. If NULL saves to current working directory.

Details

All data will be unsampled, and requests will cost money against your BigQuery quota.

Requires installation of bigQueryR and authentication under `ga_bq_auth()` or `googleAuthR::gar_auth()` with BigQuery scope set. View your projectIds upon authentication via [bqr_list_projects](#)

No segments for now.

Goals are not specified in BQ exports, so you need to look at how you define them and replicate per view e.g. unique pageviews or unique events.

Custom dimensions can be specified as session or hit level, so ignoring the setting in GA interface.

You can get a sample Google Analytics dataset in bigquery by following the instructions here: <https://support.google.com/analytics/answer/3416091?hl=en>

Value

data.frame of results

See Also

<https://support.google.com/analytics/answer/4419694?hl=en> <https://support.google.com/analytics/answer/3437719?hl=en>

google_analytics_meta *Get current dimensions and metrics available in GA API.*

Description

Get current dimensions and metrics available in GA API.

Usage

```
google_analytics_meta()
```

Value

dataframe of dimensions and metrics available to use

See Also

<https://developers.google.com/analytics/devguides/reporting/metadata/v3/reference/metadata/columns/list>

make_cohort_group *Create a cohort group*

Description

Create a cohort group

Usage

```
make_cohort_group(cohorts, lifetimeValue = FALSE, cohort_types = NULL)
```

Arguments

cohorts	A named list of start/end date pairs
lifetimeValue	lifetimeValue TRUE or FALSE. Only works for webapps.
cohort_types	placeholder, does nothing as only FIRST_VISIT_DATE supported.

Details

Example: `list("cohort 1" = c("2015-08-01", "2015-08-01"), "cohort 2" = c("2015-07-01",`

Value

A cohortGroup object

See Also

https://developers.google.com/analytics/devguides/reporting/core/v4/advanced#cohort_and_lifetime_value_ltv_dimensions_and_metrics

Other v4 cohort functions: [cohortGroup](#), [cohort_dimension_check](#), [cohort_metric_check](#), [cohort](#)

Examples

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

## authenticate,
## or use the RStudio Addin "Google API Auth" with analytics scopes set
ga_auth()

## get your accounts
account_list <- google_analytics_account_list()

## pick a profile with data to query

ga_id <- account_list[23,'viewId']

## first make a cohort group

cohort4 <- make_cohort_group(list("cohort 1" = c("2015-08-01", "2015-08-01"),
                                "cohort 2" = c("2015-07-01", "2015-07-01")))

## then call cohort report. No date_range and must include metrics and dimensions
## from the cohort list
cohort_example <- google_analytics(ga_id,
                                   dimensions=c('cohort'),
                                   cohort = cohort4,
                                   metrics = c('cohortTotalUsers'))

## End(Not run)
```

make_ga_4_req

Make a Google Analytics v4 API fetch

Description

This function constructs the Google Analytics API v4 call to be called via [fetch_google_analytics_4](#)

Usage

```
make_ga_4_req(viewId, date_range = NULL, metrics = NULL,
  dimensions = NULL, dim_filters = NULL, met_filters = NULL,
  filtersExpression = NULL, order = NULL, segments = NULL,
  pivots = NULL, cohorts = NULL, pageToken = 0, pageSize = 1000,
  samplingLevel = c("DEFAULT", "SMALL", "LARGE"), metricFormat = NULL,
  histogramBuckets = NULL)
```

Arguments

viewId	viewId of data to get.
date_range	character or date vector of format c(start, end) or for two date ranges: c(start1,end1,start2,end2)
metrics	Metric to fetch. Supports calculated metrics.
dimensions	Dimensions to fetch.
dim_filters	A filter_clause_ga4 wrapping dim_filter
met_filters	A filter_clause_ga4 wrapping met_filter
filtersExpression	A v3 API style simple filter string. Not used with other filters.
order	An order_type object
segments	List of segments as created by segment_ga4
pivots	Pivots of the data as created by pivot_ga4
cohorts	Cohorts created by make_cohort_group
pageToken	Where to start the data fetch
pageSize	How many rows to fetch. Max 100000 each batch.
samplingLevel	Sample level
metricFormat	If supplying calculated metrics, specify the metric type
histogramBuckets	For numeric dimensions such as hour, a list of buckets of data. See details in make_ga_4_req

Metrics

Metrics support calculated metrics like ga:users / ga:sessions if you supply them in a named vector.

You must supply the correct 'ga:' prefix unlike normal metrics

You can mix calculated and normal metrics like so:

```
customMetric <- c(sessionPerVisitor = "ga:sessions / ga:visitors", "bounceRate",
```

You can also optionally supply a metricFormat parameter that must be the same length as the metrics. metricFormat can be: METRIC_TYPE_UNSPECIFIED, INTEGER, FLOAT, CURRENCY, PERCENT, TIME

All metrics are currently parsed to as.numeric when in R.

Dimensions

Supply a character vector of dimensions, with or without ga: prefix.

Optionally for numeric dimension types such as ga:hour, ga:browserVersion, ga:sessionsToTransaction, etc. supply histogram buckets suitable for histogram plots.

If non-empty, we place dimension values into buckets after string to int64. Dimension values that are not the string representation of an integral value will be converted to zero. The bucket values have to be in increasing order. Each bucket is closed on the lower end, and open on the upper end. The "first" bucket includes all values less than the first boundary, the "last" bucket includes all values up to infinity. Dimension values that fall in a bucket get transformed to a new dimension value. For example, if one gives a list of "0, 1, 3, 4, 7", then we return the following buckets: -

- bucket #1: values < 0, dimension value "<0"
- bucket #2: values in [0,1), dimension value "0"
- bucket #3: values in [1,3), dimension value "1-2"
- bucket #4: values in [3,4), dimension value "3"
- bucket #5: values in [4,7), dimension value "4-6"
- bucket #6: values >= 7, dimension value "7+"

See Also

Other GAv4 fetch functions: [fetch_google_analytics_4_slow](#), [fetch_google_analytics_4](#), [google_analytics](#)

Examples

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

## authenticate,
## or use the RStudio Addin "Google API Auth" with analytics scopes set
ga_auth()

## get your accounts
account_list <- google_analytics_account_list()

## pick a profile with data to query
ga_id <- account_list[23,'viewId']

ga_req1 <- make_ga_4_req(ga_id,
  date_range = c("2015-07-30","2015-10-01"),
  dimensions=c('source','medium'),
  metrics = c('sessions'))

ga_req2 <- make_ga_4_req(ga_id,
  date_range = c("2015-07-30","2015-10-01"),
  dimensions=c('source','medium'),
  metrics = c('users'))
```

```
fetch_google_analytics_4(list(ga_req1, ga_req2))
```

```
## End(Not run)
```

meta	<i>Google Analytics API metadata</i>
------	--------------------------------------

Description

This is a local copy of the data provided by [google_analytics_meta](#)

Usage

```
meta
```

Format

A data frame with 476 rows and 15 variables

Details

Running your own call will be more up to date, but this is here in case.

It does not include the multi-channel or cohort variables.

Source

<https://developers.google.com/analytics/devguides/reporting/core/dimsmets>

met_filter	<i>Make a metric filter object</i>
------------	------------------------------------

Description

Make a metric filter object

Usage

```
met_filter(metric, operator = c("EQUAL", "LESS_THAN", "GREATER_THAN",  
  "IS_MISSING"), comparisonValue, not = FALSE)
```

Arguments

metric	metric name to filter on.
operator	How to match the dimension.
comparisonValue	What to match.
not	Logical NOT operator. Boolean.

Value

An object of class `met_fil_ga4` for use in `filter_clause_ga4`

See Also

Other filter functions: `dim_filter`, `filter_clause_ga4`

Examples

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

## authenticate,
## or use the RStudio Addin "Google API Auth" with analytics scopes set
ga_auth()

## get your accounts
account_list <- google_analytics_account_list()

## pick a profile with data to query

ga_id <- account_list[23, 'viewId']

## create filters on metrics
mf <- met_filter("bounces", "GREATER_THAN", 0)
mf2 <- met_filter("sessions", "GREATER", 2)

## create filters on dimensions
df <- dim_filter("source", "BEGINS_WITH", "1", not = TRUE)
df2 <- dim_filter("source", "BEGINS_WITH", "a", not = TRUE)

## construct filter objects
fc2 <- filter_clause_ga4(list(df, df2), operator = "AND")
fc <- filter_clause_ga4(list(mf, mf2), operator = "AND")

## make v4 request
ga_data1 <- google_analytics_4(ga_id,
                              date_range = c("2015-07-30", "2015-10-01"),
                              dimensions=c('source', 'medium'),
                              metrics = c('sessions', 'bounces'),
                              met_filters = fc,
                              dim_filters = fc2,
```

```
filtersExpression = "ga:source!=(direct)")

## End(Not run)
```

multi_select	<i>multi_select [Shiny Module]</i>
--------------	------------------------------------

Description

Shiny Module for use with [multi_selectUI](#)

Usage

```
multi_select(input, output, session, type = c("METRIC", "DIMENSION"),
             subType = c("all", "segment", "cohort"), default = NULL)
```

Arguments

input	shiny input
output	shiny output
session	shiny session
type	metric or dimension
subType	Limit selections to those relevant
default	The default selected choice. First element if NULL

Details

Call via `shiny::callModule(multi_select, "your_id")`

Value

the selected variable

See Also

Other Shiny modules: [authDropdownUI](#), [authDropdown](#), [multi_selectUI](#)

multi_selectUI	<i>multi_select UI [Shiny Module]</i>
----------------	---------------------------------------

Description

Shiny Module for use with [multi_select](#)

Usage

```
multi_selectUI(id, label = "Metric", multiple = TRUE, width = NULL)
```

Arguments

id	Shiny id
label	label
multiple	multiple select
width	width of select

Details

Create a Google Analytics variable selector

Value

Shiny UI

See Also

Other Shiny modules: [authDropdownUI](#), [authDropdown](#), [multi_select](#)

order_type	<i>Make an OrderType object</i>
------------	---------------------------------

Description

Make an OrderType object

Usage

```
order_type(field, sort_order = c("ASCENDING", "DESCENDING"),  
           orderType = c("VALUE", "DELTA", "SMART", "HISTOGRAM_BUCKET",  
                         "DIMENSION_AS_INTEGER"))
```

Arguments

field	One field to sort by
sort_order	ASCENDING or DESCENDING
orderType	Type of ordering

Details

For multiple order sorting, create separate OrderType objects to pass

Value

A order_type_ga4 object for use in GAv4 fetch

pivot_ga4	<i>Make a pivot object</i>
-----------	----------------------------

Description

Make a pivot object

Usage

```
pivot_ga4(pivot_dim, metrics, dim_filter_clause = NULL, startGroup = 1,
          maxGroupCount = 5)
```

Arguments

pivot_dim	A character vector of dimensions
metrics	Metrics to aggregate and return.
dim_filter_clause	Only data included in filter included.
startGroup	which groups of k columns are included in response.
maxGroupCount	Maximum number of groups to return.

Details

If maxGroupCount is set to -1 returns all groups.

Value

pivot object of class pivot_ga4 for use in [filter_clause_ga4](#)

Examples

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

## authenticate,
## or use the RStudio Addin "Google API Auth" with analytics scopes set
ga_auth()

## get your accounts
account_list <- google_analytics_account_list()

## pick a profile with data to query

ga_id <- account_list[23,'viewId']

## filter pivot results to
pivot_dim_filter1 <- dim_filter("medium",
                                "REGEXP",
                                "organic|social|email|cpc")

pivot_dim_clause <- filter_clause_ga4(list(pivot_dim_filter1))

pivme <- pivot_ga4("medium",
                  metrics = c("sessions"),
                  maxGroupCount = 4,
                  dim_filter_clause = pivot_dim_clause)

pivtest <- google_analytics(ga_id,
                            c("2016-01-30","2016-10-01"),
                            dimensions=c('source'),
                            metrics = c('sessions'),
                            pivots = list(pivme))

## End(Not run)
```

segmentBuilder

Create a GAv4 Segment Builder

Description

Shiny Module for use with [segmentBuilderUI](#)

Usage

```
segmentBuilder(input, output, session)
```

Arguments

input	shiny input
output	shiny output
session	shiny session

Details

Call via `shiny::callModule(segmentBuilder, "your_id")`

Value

A segment definition

Examples

```
## Not run:  
  
library(shiny)  
library(googleAnalyticsR)  
  
ui <- shinyUI(fluidPage(  
  segmentBuilderUI("test1")  
))  
  
server <- shinyServer(function(input, output, session) {  
  
  segment <- callModule(segmentBuilder, "test1")  
  
  .. use segment() in further gav4 calls.  
  
})  
  
# Run the application  
shinyApp(ui = ui, server = server)  
  
## End(Not run)
```

segmentBuilderUI

Create a GAv4 Segment Builder

Description

Shiny Module for use with [segmentBuilder](#)

Usage

```
segmentBuilderUI(id)
```

Arguments

id Shiny id

Value

Shiny UI for use in app

Examples

```
## Not run:

library(shiny)
library(googleAnalyticsR)

ui <- shinyUI(fluidPage(
  segmentBuilderUI("test1")
))

server <- shinyServer(function(input, output, session) {

  segment <- callModule(segmentBuilder, "test1")

  .. use segment() in further gav4 calls.

})

# Run the application
shinyApp(ui = ui, server = server)

## End(Not run)
```

segment_define *Make a segment definition*

Description

Defines the segment to be a set of SegmentFilters which are combined together with a logical AND operation.

segment_define is in the hierarchy of segment creation, for which you will also need:

- [segment_define](#) : AND combination of segmentFilters
- [segment_vector_simple](#) or [segment_vector_sequence](#)
- [segment_element](#) that are combined in OR lists for segment_vectors_*

Usage

```
segment_define(segment_filters, not_vector = NULL)
```

Arguments

segment_filters A list of [segment_vector_simple](#) and [segment_vector_sequence](#)

not_vector Boolean applied to each segmentFilter step. If NULL, assumed FALSE

Value

segmentDefinition object for [segment_ga4](#)

See Also

Other v4 segment functions: [segment_element](#), [segment_ga4](#), [segment_vector_sequence](#), [segment_vector_simple](#)

segment_element	<i>Make a segment element</i>
-----------------	-------------------------------

Description

segment_element is the lowest hierarchy of segment creation, for which you will also need:

- [segment_define](#) : AND combination of segmentFilters
- [segment_vector_simple](#) or [segment_vector_sequence](#)
- [segment_element](#) that are combined in OR lists for segment_vectors_*

Usage

```
segment_element(name, operator = c("REGEXP", "BEGINS_WITH", "ENDS_WITH",
  "PARTIAL", "EXACT", "IN_LIST", "NUMERIC_LESS_THAN", "NUMERIC_GREATER_THAN",
  "NUMERIC_BETWEEN", "LESS_THAN", "GREATER_THAN", "EQUAL", "BETWEEN"),
  type = c("METRIC", "DIMENSION"), not = FALSE, expressions = NULL,
  caseSensitive = NULL, minComparisonValue = NULL,
  maxComparisonValue = NULL, scope = c("SESSION", "USER", "HIT", "PRODUCT"),
  comparisonValue = NULL, matchType = c("PRECEDES", "IMMEDIATELY_PRECEDES"))
```

Arguments

name Name of the GA metric or dimension to segment on

operator How name shall operate on expression or comparisonValue

type A metric or dimension based segment element

not Should the element be the negation of what is defined

expressions [dim] What the name shall compare to

caseSensitive	[dim] Whether to be case sensitive
minComparisonValue	[dim] Minimum comparison values for BETWEEN
maxComparisonValue	Max comparison value for BETWEEN operator
scope	[met] Scope of the metric value
comparisonValue	[met] What the name shall compare to
matchType	If used in sequence segment, what behaviour

Value

An SegmentFilterClause object

See Also

Other v4 segment functions: [segment_define](#), [segment_ga4](#), [segment_vector_sequence](#), [segment_vector_simple](#)

segment_ga4	<i>Make a segment object for use</i>
-------------	--------------------------------------

Description

A Segment is a subset of the Analytics data. For example, of the entire set of users, one Segment might be users from a particular country or city.

Usage

```
segment_ga4(name, segment_id = NULL, user_segment = NULL,
            session_segment = NULL)
```

Arguments

name	The name of the segment for the reports.
segment_id	The segment ID of a built in or custom segment e.g. gaid::-3
user_segment	A list of segment_define's that apply to users
session_segment	A list of segment_define's that apply to sessions

Details

segment_ga4 is the top hierarchy of segment creation, for which you will also need:

- [segment_define](#) : AND combination of segmentFilters
- [segment_vector_simple](#) or [segment_vector_sequence](#)
- [segment_element](#) that are combined in OR lists for segment_vectors_*

Value

a segmentFilter object. You can pass a list of these to the request.

See Also

Other v4 segment functions: [segment_define](#), [segment_element](#), [segment_vector_sequence](#), [segment_vector_simple](#)

Examples

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

## authenticate,
## or use the RStudio Addin "Google API Auth" with analytics scopes set
ga_auth()

## get your accounts
account_list <- google_analytics_account_list()

## pick a profile with data to query

ga_id <- account_list[23,'viewId']

## make a segment element
se <- segment_element("sessions",
                      operator = "GREATER_THAN",
                      type = "metric",
                      comparisonValue = 1,
                      scope = "USER")

se2 <- segment_element("medium",
                      operator = "EXACT",
                      type = "dimension",
                      expressions = "organic")

## choose between segment_vector_simple or segment_vector_sequence
## Elements can be combined into clauses, which can then be
## combined into OR filter clauses
sv_simple <- segment_vector_simple(list(list(se)))

sv_simple2 <- segment_vector_simple(list(list(se2)))

## Each segment vector can then be combined into a logical AND
seg_defined <- segment_define(list(sv_simple, sv_simple2))

## if only one AND definition, you can leave out wrapper list()
```

segment_vector_sequence

Make sequenceSegment

Description

segment_vector_sequence is in the hierarchy of segment creation, for which you will also need:

- [segment_define](#) : AND combination of segmentFilters
- [segment_vector_simple](#) or [segment_vector_sequence](#)
- [segment_element](#) that are combined in OR lists for segment_vectors_*

Usage

segment_vector_sequence(segment_elements, firstStepMatch = FALSE)

Arguments

segment_elements

a list of OR lists of segment elements

firstStepMatch FALSE default

See Also

Other v4 segment functions: [segment_define](#), [segment_element](#), [segment_ga4](#), [segment_vector_simple](#)

segment_vector_simple *Make a simple segment vector*

Description

segment_vector_simple is in the hierarchy of segment creation, for which you will also need:

- [segment_define](#) : AND combination of segmentFilters
- [segment_vector_simple](#) or [segment_vector_sequence](#)
- [segment_element](#) that are combined in OR lists for segment_vectors_*

Usage

segment_vector_simple(segment_elements)

Arguments

segment_elements

A list of OR lists of [segment_element](#)

Value

A segment vector you can put in a list for use in [segment_ga4](#)

See Also

Other v4 segment functions: [segment_define](#), [segment_element](#), [segment_ga4](#), [segment_vector_sequence](#)

Index

*Topic **datasets**

- meta, [54](#)
- aggregateGADData, [3](#)
- allowed_metric_dim, [4](#)
- authDropdown, [4](#), [5](#), [56](#), [57](#)
- authDropdownUI, [4](#), [5](#), [5](#), [56](#), [57](#)
- bqr_list_projects, [49](#)
- cohort, [51](#)
- cohort_dimension_check, [51](#)
- cohort_metric_check, [51](#)
- cohortGroup, [51](#)
- dim_filter, [6](#), [9](#), [10](#), [43](#), [52](#), [55](#)
- fetch_google_analytics_4, [7](#), [9](#), [45](#), [51](#), [53](#)
- fetch_google_analytics_4_slow, [8](#), [8](#), [45](#), [53](#)
- filter_clause_ga4, [6](#), [9](#), [43](#), [52](#), [55](#), [58](#)
- ga_account_list
 - (google_analytics_account_list), [48](#)
- ga_accounts, [10](#), [12](#), [19–23](#), [25–28](#), [30–37](#), [39–41](#), [48](#)
- ga_adwords, [11](#), [11](#), [12](#), [19–23](#), [25–28](#), [30–37](#), [39–41](#), [48](#)
- ga_adwords_list, [11](#), [12](#), [12](#), [19–23](#), [25–28](#), [30–37](#), [39–41](#), [48](#)
- ga_auth, [13](#)
- ga_cache_call, [14](#)
- ga_custom_datasource, [15](#), [16](#), [17](#), [19](#)
- ga_custom_upload, [15](#), [15](#), [17](#), [19](#)
- ga_custom_upload_file, [15](#), [16](#), [17](#), [19](#)
- ga_custom_upload_list, [15–17](#), [18](#)
- ga_custom_vars, [11](#), [12](#), [19](#), [20–23](#), [25–28](#), [30–37](#), [39–41](#), [48](#)
- ga_custom_vars_list, [11](#), [12](#), [19](#), [20](#), [21–23](#), [25–28](#), [30–37](#), [39–41](#), [48](#)
- ga_experiment, [11](#), [12](#), [19](#), [20](#), [20](#), [21–23](#), [25–28](#), [30–37](#), [39–41](#), [48](#)
- ga_experiment_list, [11](#), [12](#), [19–21](#), [21](#), [22](#), [23](#), [25–28](#), [30–37](#), [39–41](#), [48](#)
- ga_filter, [11](#), [12](#), [19–21](#), [22](#), [23](#), [25–27](#), [29–37](#), [39–41](#), [48](#)
- ga_filter_add, [11](#), [12](#), [19–22](#), [22](#), [25–28](#), [30–37](#), [39–41](#), [48](#)
- ga_filter_apply_to_view, [11](#), [12](#), [19–23](#), [24](#), [26–28](#), [30–37](#), [39–41](#), [48](#)
- ga_filter_delete, [11](#), [12](#), [19–23](#), [25](#), [25](#), [26](#), [27](#), [29–37](#), [39–41](#), [48](#)
- ga_filter_list, [11](#), [12](#), [19–23](#), [25](#), [26](#), [26](#), [27](#), [29–37](#), [39–41](#), [48](#)
- ga_filter_update, [11](#), [12](#), [19–23](#), [25](#), [26](#), [27](#), [29–37](#), [39–41](#), [48](#)
- ga_filter_update_filter_link, [11](#), [12](#), [19–23](#), [25–27](#), [28](#), [30–37](#), [39–41](#), [48](#)
- ga_filter_view, [11](#), [12](#), [19–23](#), [25–27](#), [29](#), [29](#), [31–37](#), [39–41](#), [48](#)
- ga_filter_view_list, [11](#), [12](#), [19–23](#), [25–27](#), [29](#), [30](#), [30](#), [31–37](#), [39–41](#), [48](#)
- ga_goal, [11](#), [12](#), [19–23](#), [25–27](#), [29–31](#), [31](#), [32–37](#), [39–41](#), [48](#)
- ga_goal_list, [11](#), [12](#), [19–23](#), [25–27](#), [29–31](#), [32](#), [33–37](#), [39–41](#), [48](#)
- ga_remarketing_estimate, [11](#), [12](#), [19–23](#), [25–27](#), [29–32](#), [32](#), [34–37](#), [39–41](#), [48](#)
- ga_remarketing_get, [11](#), [12](#), [19–23](#), [25–27](#), [29–33](#), [33](#), [34–37](#), [39–41](#), [48](#)
- ga_remarketing_list, [11](#), [12](#), [19–23](#), [25–27](#), [29–34](#), [34](#), [35–37](#), [39–41](#), [48](#)
- ga_segment_list, [11](#), [12](#), [19–23](#), [25–27](#), [29–34](#), [35](#), [36](#), [37](#), [39–41](#), [48](#)
- ga_unsampled, [11](#), [12](#), [19–23](#), [25–27](#), [29–35](#), [35](#), [37](#), [39–41](#), [48](#)
- ga_unsampled_download, [36](#)
- ga_unsampled_list, [11](#), [12](#), [19–23](#), [25–27](#), [29–36](#), [37](#), [39–41](#), [48](#)

ga_users_list, [11](#), [12](#), [19–23](#), [25–27](#), [29–37](#),
[38](#), [39–41](#), [48](#)
ga_view, [11](#), [12](#), [19–23](#), [25–27](#), [29–37](#), [39](#), [39](#),
[40](#), [41](#), [48](#)
ga_view_list, [11](#), [12](#), [19–23](#), [25–27](#), [29–37](#),
[39](#), [40](#), [41](#), [48](#)
ga_webproperty, [11](#), [12](#), [19–23](#), [25–27](#),
[29–37](#), [39](#), [40](#), [40](#), [41](#), [48](#)
ga_webproperty_list, [11](#), [12](#), [19–23](#), [25–27](#),
[29–37](#), [39–41](#), [41](#), [48](#)
gar_auth, [13](#), [14](#)
gar_auth_service, [13](#)
gar_cache_empty, [44](#)
gar_cache_setup, [44](#)
gar_set_client, [44](#)
getColNameOfClass, [42](#)
google_analytics, [8](#), [9](#), [43](#), [44](#), [45](#), [53](#)
google_analytics_3, [43](#), [45](#)
google_analytics_4 (google_analytics),
[43](#)
google_analytics_account_list, [11](#), [12](#),
[19–23](#), [25–27](#), [29–37](#), [39–41](#), [48](#)
google_analytics_bq, [48](#)
google_analytics_meta, [50](#), [54](#)
googleAnalyticsR, [42](#)
googleAnalyticsR-package
(googleAnalyticsR), [42](#)

make_cohort_group, [43](#), [50](#), [52](#)
make_ga_4_req, [7–9](#), [43](#), [45](#), [51](#), [52](#)
met_filter, [6](#), [9](#), [10](#), [43](#), [52](#), [54](#)
meta, [54](#)
multi_select, [5](#), [56](#), [57](#)
multi_selectUI, [5](#), [56](#), [57](#)

order_type, [43](#), [52](#), [57](#)

pivot_ga4, [43](#), [52](#), [58](#)

segment_define, [61](#), [61](#), [62–64](#), [66](#), [67](#)
segment_element, [61](#), [62](#), [62](#), [63](#), [64](#), [66](#), [67](#)
segment_ga4, [43](#), [52](#), [62](#), [63](#), [63](#), [66](#), [67](#)
segment_vector_sequence, [61–64](#), [66](#), [66](#),
[67](#)
segment_vector_simple, [61–64](#), [66](#), [66](#)
segmentBuilder, [59](#), [60](#)
segmentBuilderUI, [59](#), [60](#)
Startup, [13](#)
Sys.setenv, [13](#)