

Package ‘RGoogleAnalytics’

October 9, 2018

Version 0.1.6

Date 2018-10-01

Title R Wrapper for the Google Analytics API

Author Michael Pearmain. Contributions from Nick Mihailowski,
Vignesh Prajapati, Kushan Shah, Nicolas Remy and Manuel de Francisco Vera
person("Manuel", "de Francisco", email = "mdefraver@hotelspecials.nl", role = c("ctb"))

Description Provides functions for accessing and retrieving data from the
Google Analytics API.

Maintainer Manuel de Francisco Vera <manueldefrancisco@gmail.com>

BugReports <https://github.com/ManuelDeFrancisco/RGoogleAnalytics/issues>

Depends R(>= 3.0.2), lubridate, httr, digest (>= 0.6.8)

Suggests testthat

License Apache License 2.0

NeedsCompilation no

Repository CRAN

RoxygenNote 6.0.1

Date/Publication 2018-10-09 15:30:02 UTC

R topics documented:

Auth	2
GetProfiles	3
GetReportData	3
Init	5
ValidateToken	6
Index	7

Auth	<i>Authorize the RGoogleAnalytics package to the user's Google Analytics Account using OAuth2.0</i>
------	---

Description

This function expects a Client ID and Client Secret. In order to obtain these, you will have to register an application with the Google Analytics API. This can be done as follows

- Go to <https://console.developers.google.com>
- Create a New Project and enable the Google Analytics API
- On the Credentials screen, create a new Client ID for Application Type "Installed Application".
- Copy the Client ID and Client Secret to your R Script as shown in the Examples section below

Usage

```
Auth(client.id, client.secret)
```

Arguments

<code>client.id</code>	Equivalent to a user name
<code>client.secret</code>	Equivalent to a password

Details

When evaluated for the first time this function asks for User Consent for the Google Analytics Account and creates a OAuth Token Object. The token object can be saved locally to a file on the user's system. In subsequent runs, User Consent is not required unless you are querying a Google Analytics profile associated with a different email account. This function uses `oauth2.0_token` under the hood to create the OAuth Tokens. The Access Token has a 60 minute lifetime after which it expires and a new token has to be obtained. This can be done using the `ValidateToken` method

Value

`google.token` A Token object containing all the data required for OAuth access. See `Token2.0` for additional information on the Token object

Examples

```
## Not run:  
# Generate the oauth_token object  
oauth_token <- Auth(client.id = "150487456763-XXXXXXXXXXXXXXXX.apps.googleusercontent.com",  
client.secret = "TUXXXXXXXXXXXXXX_TknUI")  
  
# Save the token object for future sessions  
save(oauth_token, file="oauth_token")  
  
# Load the token object
```

```
load("oauth_token")  
## End(Not run)
```

GetProfiles	<i>Retrieve the list of Profiles for the Google Analytics Account</i>
-------------	---

Description

This function retrieves all the available profiles from the Google analytics account. Before retrieving the list of profiles, this function checks whether the token is valid and regenerates it if required

Usage

```
GetProfiles(token)
```

Arguments

token Token Object generated by [Auth](#)

Value

R dataframe with profile ID and profile name.

GetReportData	<i>Query the Google Analytics API for the specified dimensions, metrics and other query parameters</i>
---------------	--

Description

This function will retrieve the data by firing the query to the Core Reporting API. It also displays status messages after the completion of the query. The user also has the option split the query into daywise partitions and paginate the query responses in order to decrease the effect the sampling

Usage

```
GetReportData(query.builder, token, split_daywise = FALSE,  
              paginate_query = FALSE, delay = 0)
```

Arguments

query.builder	Name of the object created using QueryBuilder
token	Name of the token object created using Auth
split_daywise	Splits the query by date range into sub queries of single days. Setting this argument to True automatically paginates through each daywise query. Note that if this argument is set to True, queries will take more longer to complete and use more Quota
paginate_query	Pages through chunks of results by requesting maximum number of allowed rows at a time. Note that if this argument is set to True, queries will take more longer to complete and use more Quota. For more on Google Analytics API Quota check https://developers.google.com/analytics/devguides/reporting/core/v3/limits-quotas#core_reporting
delay	Since Pagination and Query splitting fire successive queries, there is a possibility of getting Quota Error: Rate Limit Exceeded from the Google Analytics API. This parameter can be used to specify a Time delay (in seconds) between successive queries in order to stay within the Google Analytics API Rate Limits

Value

dataframe containing the response from the Google Analytics API

See Also

Prior to executing the query, as a good practice queries can be tested in the Google Analytics Query Feed Explorer at <http://ga-dev-tools.appspot.com/explorer/>

Examples

```
## Not run:
# This example assumes that a token object is already created

# Create a list of Query Parameters
query.list <- Init(start.date = "2014-11-28",
                  end.date = "2014-12-04",
                  dimensions = "ga:date",
                  metrics = "ga:sessions,ga:pageviews",
                  max.results = 1000,
                  table.id = "ga:33093633")

# Create the query object
ga.query <- QueryBuilder(query.list)

# Fire the query to the Google Analytics API
ga.df <- GetReportData(query, oauth_token)
ga.df <- GetReportData(query, oauth_token, split_daywise=True)
ga.df <- GetReportData(query, oauth_token, paginate_query=True)

## End(Not run)
```

Init	<i>Initialize the Google Analytics query parameters</i>
------	---

Description

This function takes all the query parameters and combines them into a single list that is to be passed as an argument to [QueryBuilder](#). Note that parameter validation is performed when the [QueryBuilder](#) object is created

Usage

```
Init(start.date = NULL, end.date = NULL, dimensions = NULL,
      metrics = NULL, filters = NULL, sort = NULL, segments = NULL,
      max.results = NULL, start.index = NULL, table.id = NULL,
      caching.dir = NULL, caching = FALSE)
```

Arguments

start.date	Start Date for fetching Analytics Data. Start Date must be of the format "%Y-%m-%d"
end.date	End Date for fetching Analytics Data. End Date must be of the format "%Y-%m-%d"
dimensions	Optional. A vector of up to 7 dimensions, either as a single string or a vector or strings, E.g. "ga:source,ga:medium" or c("ga:source", "ga:medium").
metrics	A vector of up to 10 metrics, either as a single string or a vector or strings. E.g. "ga:sessions" or c("ga:sessions", "ga:bounces").
filters	Optional.The filter string for the GA request.e.g. "ga:medium==referral".
sort	Optional.The sorting order for the data to be returned.e.g. "ga:sessions" or c("ga:sessions", "-ga:browser")
segments	Optional.An advanced segment definition to slice and dice your Analytics data.
max.results	Optional.Maximum Number of rows to include in the query response. Default value is 10000
start.index	Optional.The first row of data to retrieve. Default value is 1
table.id	Profile ID of the form ga:XXXXXX where XXXXXX is the Analytics View (Profile) ID of for which the query will retrieve the data. The View ID can be found under View Settings by navigating to the Admin Tab under your Google Analytics Profile
caching.dir	String Directory to save cached data
caching	Boolean caching required?

Value

List of all the Query Parameters initialized by the user

See Also

Valid Combinations of Dimensions and Metrics can be found at <http://code.google.com/apis/analytics/docs/gdata/gdataReferenceDimensionsMetrics.html#validCombinations>

ValidateToken	<i>Check whether the Access Token has expired</i>
---------------	---

Description

This function checks whether the Access Token is expired. If yes, it generates a new Access Token and updates the token object.

Usage

```
ValidateToken(token)
```

Arguments

token	Token object containing the OAuth authentication parameters
-------	---

Index

Auth, [2](#), [3](#), [4](#)

GetProfiles, [3](#)

GetReportData, [3](#)

Init, [5](#)

oauth2.0_token, [2](#)

QueryBuilder, [4](#), [5](#)

Token2.0, [2](#)

ValidateToken, [2](#), [6](#)