Package ‘RGoogleAnalyticsPremium’

October 12, 2022

Type Package

Title Unsampled Data in R for Google Analytics Premium Accounts

Version 0.1.1

Date 2015-10-21

Author Jalpa Joshi Dave

Maintainer Jalpa Joshi Dave <jalpa@tatvic.com>

Description It fires a query to the API to get the unsampled data in R for Google Analytics Premium Accounts. It retrieves data from the Google drive document and stores it into the local drive. The path to the excel file is returned by this package. The user can read data from the excel file into R using read.csv() function.

License Apache License 2.0

Depends R(>= 3.2.2), httr, lubridate, jsonlite, utils

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2015-11-02 11:51:18

R topics documented:

`Auth` *Authorize the RGoogleAnalytics package to the user's Google Analytics Account using OAuth2.0*

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, Contacts API and Drive 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

`client.id`	Equivalent to a user name
`client.secret`	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-XXXXXXXXXXXXXXX.apps.googleusercontent.com",
client.secret = "TUXXXXXXXXXXXX_TknUI")

# Save the token object for future sessions
save(oauth_token, file="oauth_token")

# Load the token object
```

```
load("oauth_token")

## End(Not run)
```

`GetFile` *Query the Google Analytics Premium API for the specified dimensions, metrics and other query parameters*

Description

Query the Google Analytics Premium API for the specified dimensions, metrics and other query parameters

Usage

```
GetFile(query.builder, token, accountid, webpropertyid, profileid)
```

Arguments

`query.builder`	Name of the object created using `QueryBuilder`
`token`	Name of the token object created using `Auth`
`accountid`	Google analytics premium account id
`webpropertyid`	Webproperty ID for google analytics premium account
`profileid`	View ID for google analytics premium account

Value

It returns path to file on local drive that contains extracted unsampled data.

`Init` *Initialize the Google Analytics query parameters*

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(end.date = NULL, metrics = NULL, start.date = NULL, title = NULL,
  dimensions = NULL, filters = NULL, segments = NULL)
```

Arguments

`end.date`	End Date for fetching Analytics Data. End Date must be of the format "%Y-%m-%d"
`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").
`start.date`	Start Date for fetching Analytics Data. Start Date must be of the format "%Y-%m-%d"
`title`	Title of unsampled report.
`dimensions`	Optional. A vector of up to 4 dimensions, either as a single string or a vector or strings, E.g. "ga:source,ga:medium" or c("ga:source", "ga:medium").
`filters`	Optional.The filter string for the GA request.e.g. "ga:medium==referral".
`segments`	Optional.An advanced segment definition to slice and dice your Analytics data.

Value

List of all the Query Parameters initialized by the user

`QueryBuilder` *Initialize a QueryBuilder object with the given parameters and perform validation*

Description

Initialize a QueryBuilder object with the given parameters and perform validation

Usage

```
QueryBuilder(query.params.list)
```

Arguments

`query.params.list`	List of all the Query Parameters. See `Init` for the entire list

Value

The builder object to process the query parameters.

ToBody — *Returns the URI constructed from the parameter settings. This also URI-encodes all the values in each query parameter.*

Description

Returns the URI constructed from the parameter settings. This also URI-encodes all the values in each query parameter.

Usage

```
ToBody(query.builder, token)
```

Arguments

`query.builder`	Name of the Object of the Query Builder Class
`token`	Token object containing the OAuth2.0 Authentication details

Value

A full URI that can be used with the Google Analytics API.

ValidateToken — *Check whether the Access Token has expired*

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