Package 'RAdwords'

January 20, 2025

Type Package

Title Loading Google Adwords Data into R

Description Aims at loading Google Adwords data into R. Adwords is an online advertising service that enables advertisers to display advertising copy to web users (see <https://developers.google.com/adwords/> for more information). Therefore the package implements three main features. First, the package provides an authentication process for R with the Google Adwords API (see <https://developers.google.com/adwords/api/> for more information) via OAUTH2. Second, the package offers an interface to apply the Adwords query language in R and query the Adwords API with ad-hoc reports. Third, the received data are transformed into suitable data formats for further data processing and data analysis.

Version 0.1.18

Author Johannes Burkhardt <johannes.burkhardt@gmail.com>, Matthias Bannert <matthias.bannert@gmail.com>

Maintainer Johannes Burkhardt <johannes.burkhardt@gmail.com>

Depends R (>= 3.0.0)

Imports RCurl, rjson

Suggests testthat

License MIT + file LICENSE

URL https://github.com/jburkhardt/RAdwords, https://developers.google.com/adwords, https://developers.google.com/adwords/api/

BugReports https://github.com/jburkhardt/RAdwords/issues

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-28 10:20:03 UTC

Contents

`changeNames` *Change Names of Attributes/variables*

Description

Converts the default display names into nicer or more practical names.

Usage

```
changeNames(data)
```

Arguments

`data`	Transformed dataframe

Value

New column names of dataframe.

`doAuth` *Invoke the Authentication Process with Google*

Description

This function starts the authentication process with Google. Note that this functions needs user interaction.

Usage

```
doAuth(save = T)
```

Arguments

save	logical denotes whether authentication information should be saved on disk. Defaults to TRUE.

See Also

getAuth,loadToken

---

getAuth *Authentication of R app*

Description

getAuth authenticates the R app at the Google authentication server using OAUTH2 and receives the client token. Usually you need not to run getAuth() explicitly since the whole authentication process is managed by doAuth.

Usage

```
getAuth()
```

Value

Client token from Google authentication server. Dataframe with the credential information which is cached in working space and optionally saved as RData file in current working directory.

---

getData *Get Adwords Data*

Description

getData posts the Adwords Query Language (awql) Statement which is generated with statement. The data are retrieved from the Adwords API as a dataframe.

Usage

```
getData(clientCustomerId, google_auth, statement, apiVersion = "201809",
  transformation = TRUE, changeNames = TRUE,
  includeZeroImpressions = FALSE, verbose = FALSE)
```

Arguments

clientCustomerId	Adwords Account Id
google_auth	list of authentication
statement	awql statement generated with statement.
apiVersion	supports 201809, 201806, 201802 defaults to 201806.
transformation	If TRUE, data will be transformed with transformData into suitable R dataframe. Else, the data are returned in raw format.
changeNames	If TRUE, the display names of the transformed data are converted into more nicer/practical names. Requires transformation = TRUE
includeZeroImpressions	If TRUE zero impressions will be included. Defaults to FALSE.
verbose	Defaults to FALSE. If TRUE, the curl connection output will be printed.

Value

Dataframe with the Adwords Data.

---

loadToken *Loading the Access Token*

Description

loadToken loads the access token using credentials provided by getAuth. Execution of function is possible only once per authentication process. Usually you need not to run loadToken() explicitly since the whole authentication process is managed by doAuth.

Usage

```
loadToken(credlist)
```

Arguments

credlist	list of credentials

Value

Access token with corresponding time stamp.

metrics *Get Metrics/Attributes of specified Report*

Description

metrics provides an overview of all available metrics/attributes for a specified report type.

Usage

```
metrics(report = "ACCOUNT_PERFORMANCE_REPORT", apiVersion = "201809")
```

Arguments

report	Report type
apiVersion	Supports 201809, 201806, 201802. Defaults to 201809.

Value

List of available metrics/attributes.

RAdwords *Loading Google Adwords Data into R*

Description

The aim of **RAdwords** is loading Google Adwords data into R. Therefore the package implements three main features.
First, the package provides an **authentication process** for **R** with the **Adwords API** via OAUTH2.
Second, the package offers an interface to apply the Adwords query language in R and **query the Adwords API** with ad-hoc reports.
Third, the received **data are transformed into suitable data format** for further data processing and data analysis.

Details

Requirements:
In order to access the Adwords API you have to set up a Google API project for native apps. The Google API project provides a **Client Id** and **Client Secret** which is necessary for the authentication. Moreover you need to have a Adwords MCC (My Client Center) with an **Adwords developer token**.

Authentication:
doAuth manages the complete authentication process. Meaning doAuth authenticates the R app for the first time, loads the access token or refreshes the access token if expired. Hence, you only run doAuth() to authenticate whether it is your initial R Session or a later instance.

What's happening in details?
Once the API projects for native application is set up, getAuth is able to authenticate the R app with the credentials (Client Id, Client Secret) from the Google API project. The Google authentication server returns a client token, which later is used by loadToken to receive the access token. If the access token is expired after one hour, it can be updated with refreshToken. The access token in combination with the Adwords developer token enables a connection with the Adwords API.

Create Statement:
statement creates the Adwords Query Language Statement.

Receiving Data:
getData queries the data from the Adwords API and transforms the data into an R dataframe.

Author(s)

Johannes Burkhardt <johannes.burkhardt@gmail.com>
Matthias Bannert <matthias.bannert@gmail.com>
https://github.com/jburkhardt/RAdwords

Examples

```
## Not run:
Authentication:
google_auth <- doAuth()
Create Statement:
body <- statement(select = c('Clicks','AveragePosition','Cost','Ctr'),
                  report = "ACCOUNT_PERFORMANCE_REPORT",
                  start = "2018-01-01",
                  end = "2018-01-10")
Query Adwords API and load data as dataframe:
data <- getData(clientCustomerId = 'xxx-xxx-xxxx', #use Adwords Account Id (MCC Id will not work)
                google_auth = google_auth,
                statement = body)
Get available report types:
reports()
Get available metrics/attributes of specific report type:
metrics(report = 'ACCOUNT_PERFORMANCE_REPORT')

## End(Not run)
```

---

refreshToken *Refresh Access Token*

Description

refreshToken returns a new valid access token. The access token deprecates after one hour and has to updated with the refresh token. Usually you need not to run refreshToken() explicitly since the whole authentication process is managed by doAuth.

Usage

```
refreshToken(google_auth)
```

Arguments

google_auth	list of credentials and access token

Value

New access token with corresponding time stamp.

---

reports *Show available Adwords Reports*

Description

reports provides an overview of all available Adwords report types. The report type is specified in statement.

Usage

```
reports(apiVersion = "201809")
```

Arguments

apiVersion	Supports 201809, 201806 and 201802. Defaults to 201809.

Value

Available report types.

---

statement *Build Adwords Query Language Statement*

Description

Generates and builds the Adwords Query Language Statement for querying the Adwords API.

Usage

```
statement(select = c("AccountDescriptiveName", "AccountId", "Impressions",
  "Clicks", "Cost", "Date"), report = "ACCOUNT_PERFORMANCE_REPORT", where,
  start = "2018-01-01", end = "2018-01-10", apiVersion = "201809",
  compress = TRUE)
```

Arguments

select	Attributes
report	Report type
where	Condition list, e.g. "CampaignName STARTS_WITH 'A' AND Clicks > 100", multiple conditions can be only combined with AND Operators: = | != | > | >= | < | <= | IN | NOT_IN | STARTS_WITH | STARTS_WITH_IGNORE_CASE | CONTAINS | CONTAINS_IGNORE_CASE | DOES_NOT_CONTAIN | DOES_NOT_CONTAIN_IGNC
start	Beginning of date range. Format: 2018-01-01
end	End of date rage. Format: 2018-01-10
apiVersion	Adwords API Version, supports 201809, 201806, 201802 defaults to 201809.
compress	TRUE / FALSE, Gzipped data download if TRUE

Value

The statement neccessary for the getData function.

Examples

```
body <- statement(select=c('CampaignName','Clicks','Cost','Ctr'),
                  report="CAMPAIGN_PERFORMANCE_REPORT",
                  where="CampaignName STARTS_WITH 'A' AND Clicks > 100",
                  start="2018-01-20",
                  end="2018-01-21")
body <- statement(select=c('Criteria','Clicks','Cost','Ctr'),
                  report="KEYWORDS_PERFORMANCE_REPORT",
                  where="Clicks > 100",
                  start="2018-01-20",
                  end="2018-01-21")
body <- statement(select=c('Clicks','AveragePosition','Cost','Ctr'),
                  report="ACCOUNT_PERFORMANCE_REPORT",
                  start="2018-01-20",
                  end="2018-01-21")
```

---

transformData *Transform data into R dataframe*

Description

Transforms the csv data file received from the Adwords API into a dataframe. Moreover the variables are converted into suitable formats. The function is used inside getData and parameters are set automatically.

Usage

```
transformData(data, report = reportType, apiVersion = "201809")
```

Arguments

`data`	Raw csv data from Adwords API.
`report`	Report type.
`apiVersion`	set automatically by `getData`. Supported are 201809, 201806, 201802. Defaults to 201809.

Value

Dataframe with the Adwords Data.

Index