

Package: rfacebookstat (via r-universe)

October 15, 2026

Type Package

Title Load Data from Facebook API Marketing

Version 2.18.0

Maintainer Alexey Seleznev <selesnow@gmail.com>

Description Load data by campaigns, ads, ad sets and insights, ad account and business manager from Facebook Marketing API into R. For more details see official documents by Facebook Marketing API
<<https://developers.facebook.com/documentation/ads-commerce/marketing-api>>.

Depends R (>= 4.1.0)

BugReports <https://github.com/selesnow/rfacebookstat/issues>

Imports httr, httr2, jsonlite, dplyr, purrr, tidyr (>= 1.0.0), stringr, tidyselect, pbapply

URL <https://selesnow.github.io/rfacebookstat/>,
https://www.youtube.com/playlist?list=PLD2LDq8edf4pIt0b-vZTG5AXZK2niJ8_R

License MIT + file LICENSE

Suggests knitr, rmarkdown

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libssl-dev

Repository <https://selesnow.r-universe.dev>

Date/Publication 2026-10-15 18:16:48 UTC

RemoteUrl <https://github.com/selesnow/rfacebookstat>

RemoteRef HEAD

RemoteSha 03e58861eb7ebf523f72b9f7adbbecca5e0899879

Contents

rfacebookstat-package	2
fbAuth	3
fbDeleteAdAccountUsers	4
fbDownloadBusinessInvoices	6
fbGetAdAccount	7
fbGetAdAccountCustomAudiences	8
fbGetAdAccounts	9
fbGetAdAccountsConversions	11
fbGetAdAccountUsers	12
fbGetAdAccountUsersPermissions	13
fbGetAdCreative	14
fbGetAds	15
fbGetAdSets	17
fbGetAdVideos	19
fbGetApps	20
fbGetBusinessInvoices	21
fbGetBusinessManagers	22
fbGetBusinessManagersUsers	23
fbGetBusinessUserAdAccounts	24
fbGetCampaigns	25
fbGetCatalogs	26
fbGetCostData	27
fbGetLogins	28
fbGetLongTimeToken	29
fbGetMarketingStat	29
fbGetPages	35
fbGetSettings	36
fbGetToken	36
fbGetUserAdAccounts	37
fbSetters	38
fbUpdateAdAccountUsers	38
Index	40

rfacebookstat-package *Load Data from Facebook API Marketing*

Description

Load data by campaigns, ads, ad sets and insights, ad account and business manager from Facebook Marketing API into R. For more details see official documents by Facebook Marketing API <<https://developers.facebook.com/documentation/ads-commerce/marketing-api>>.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

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

Maintainer: Alexey Seleznev <selesnow@gmail.com>

See Also

See official documentation: rfacebookstat package - <https://selesnow.github.io/rfacebookstat/> Facebook API Marketing - <https://developers.facebook.com/documentation/ads-commerce/marketing-api>

Examples

```
## Not run:
library(rfacebookstat)
# Auth
token <- fbGetToken(app_id = 000000000000000)

# Get statistic
CampStat <- fbGetMarketingStat(accounts_id = "act_0000000000",
  level = "campaign", fields = "campaign_name,impressions,clicks",
  breakdowns = "age", sorting = "unique_impressions_descending",
  filtering = "[{'field':'age','operator':'IN','value':['18-24','25-34']}]",
  date_start = "2016-10-01", date_stop = "2016-10-10",
  access_token = token)

## End(Not run)
```

fbAuth

Authorization in Facebook.

Description

Wrapper for fbGetToken() and fbGetLongtimeToken().

Usage

```
fbAuth(username    = getOption("rfacebookstat.username"),
        app_id     = getOption("rfacebookstat.app_id"),
        app_secret  = getOption("rfacebookstat.app_secret"),
        token_path  = fbTokenPath(),
        scopes      = c("ads_read",
```

```

        "business_management",
        "pages_manage_ads",
        "ads_management",
        "public_profile"),
    reauth      = FALSE,
    skip_option = FALSE)

```

Arguments

username	your username on Facebook
app_id	ID of Facebook App
app_secret	Secret of Facebook App
token_path	path to dir with credentials
scopes	Permissions provide a way for your app to access data from Facebook. For detail see documentation
reauth	boolean, do reauthorization
skip_option	logical, skip access token from options or not

Value

API token object

Author(s)

Alexey Seleznev

Examples

```

## Not run:
fbAuth()

## End(Not run)

```

fbDeleteAdAccountUsers

Delete users from ad accounts.

Description

fbDeleteAdAccountUsers is method for remove users access from ad accounts.

Usage

```
fbDeleteAdAccountUsers(  
  user_ids      = NULL,  
  accounts_id   = getOption("rfacebookstat.accounts_id"),  
  api_version   = getOption("rfacebookstat.api_version"),  
  username      = getOption("rfacebookstat.username"),  
  token_path    = fbTokenPath(),  
  access_token  = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	vector with ID of your ad account, for example c("act_00001","act_00002").
access_token	Your facebook API token.
user_ids	vector with users ids who need remove from ad account.
username	your username on Facebook
token_path	path to dir with credentials
api_version	Current Facebook API version.

Details

ads_management permission is required. You need 'Administrator' access to the ad account to remove users

Author(s)

Alexey Seleznev

Examples

```
## Not run:  
#Attach package  
library(rfacebookstat)  
#Get token, where 00000000000000 is id of your app in facebook  
tk <- fbAuth(00000000000000)  
#Remove user with id 001 and 002 from accounts act_0001 and act_0002  
fbDeleteAdAccountUsers(user_ids = c(001,002),  
                        accounts_id = c("act_0001","act_0002"),  
                        api_version = "v2.12",  
                        access_token = "abcdef123456" )  
  
## End(Not run)
```

`fbDownloadBusinessInvoices`*Download Business Invoices*

Description

`fbDownloadBusinessInvoices` downloads invoice PDF files from a vector of URLs. It automatically handles both Graph API URLs (adding access tokens) and direct CDN URLs.

Usage

```
fbDownloadBusinessInvoices(  
  urls,  
  dest_dir = ".",  
  file_names = NULL,  
  username = getOption("rfacebookstat.username"),  
  token_path = fbTokenPath(),  
  access_token = getOption("rfacebookstat.access_token")  
)
```

Arguments

<code>urls</code>	A character vector of URLs (from 'download_uri' or 'cdn_download_uri').
<code>dest_dir</code>	Destination directory where files will be saved. Defaults to the current working directory.
<code>file_names</code>	Optional character vector of file names. If NULL, names will be auto-generated.
<code>username</code>	your username on Facebook
<code>token_path</code>	path to dir with credentials
<code>access_token</code>	Your facebook API token

Value

A data frame (tibble) containing the download status for each URL.

Author(s)

Alexey Seleznev

fbGetAdAccount	<i>Get ad account details by ID.</i>
----------------	--------------------------------------

Description

Load ad account details for a specific ad account or a vector of ad account IDs from Facebook Marketing API into R.

Usage

```
fbGetAdAccount(
  account_id = getOption("rfacebookstat.account_id"),
  fields = paste0("id,name,account_id,account_status,amount_spent,",
                  "created_time,balance,business_name,currency,owner,",
                  "timezone_name,disable_reason,spend_cap"),
  api_version = getOption("rfacebookstat.api_version"),
  username = getOption("rfacebookstat.username"),
  token_path = fbTokenPath(),
  access_token = getOption("rfacebookstat.access_token")
)
```

Arguments

account_id	Vector of one or more Facebook Ad Account IDs (with or without 'act_' prefix).
fields	Character string or vector of requested fields separated by comma. Default includes basic account info, status, spend and balance.
api_version	Current Facebook Marketing API version.
username	Your username on Facebook.
token_path	Path to directory with credentials.
access_token	Your Facebook API access token.

Value

Data frame (tibble) with ad account details.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
# Get details for a single account
acc <- fbGetAdAccount(account_id = "1090055510425474")

# Get details for multiple accounts
```

```
accs <- fbGetAdAccount(account_id = c("1090055510425474", "act_2090055510425474"))

## End(Not run)
```

fbGetAdAccountCustomAudiences

The Custom Audiences Associated With the Ad Account.

Description

Get The custom audiences associated with the ad account.

Usage

```
fbGetAdAccountCustomAudiences(
  business_ids = getOption("rfacebookstat.business_id"),
  accounts_id = getOption("rfacebookstat.accounts_id"),
  pixel_id = NULL,
  filtering = NULL,
  api_version = getOption("rfacebookstat.api_version"),
  username = getOption("rfacebookstat.username"),
  token_path = fbTokenPath(),
  access_token = getOption("rfacebookstat.access_token")
)
```

Arguments

business_ids	Your business manager or project ID.
accounts_id	Your account ID, for example 'act_1111111111111111'.
pixel_id	This param fetches audiences associated to specific pixel.
filtering	Filters on the report data. This parameter is an array of filter objects.
api_version	Current Facebook API version.
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token.

Value

data.frame with custom audiences data

Author(s)

Alexey Seleznev

See Also

[Ad Account Customaudiences API Documentation](#)

fbGetAdAccounts	<i>Get available ad account list.</i>
-----------------	---------------------------------------

Description

Load data frame with ad accounts available in your Business Manager or user profile from Facebook Marketing API into R.

Usage

```
fbGetAdAccounts(
  source_id = getOption("rfacebookstat.business_id"),
  api_version = getOption("rfacebookstat.api_version"),
  username = getOption("rfacebookstat.username"),
  token_path = fbTokenPath(),
  access_token = getOption("rfacebookstat.access_token"),
  limit = 50,
  filtering = NULL
)
```

Arguments

source_id	Your Business Manager or project ID. If NULL, accounts for current user are returned ('me/adaccounts').
access_token	Your Facebook API access token.
username	Your username on Facebook.
token_path	Path to directory with credentials.
api_version	Current Facebook Marketing API version.
limit	Number of ad accounts requested per API call. Reduce this if you experience unexpected errors. Default is 50.
filtering	Filter ad accounts at the API level. Can be passed as an R list, a JSON string, or a character vector shortcut. See Details.

Details

The filtering argument allows filtering ad accounts directly on the Facebook API level, reducing the volume of data transferred and speeding up execution.

Filters can be specified in three formats:

- **R list:** A list of lists, where each sublist has elements field, operator, and value.
- **JSON string:** A JSON array of filter objects, e.g., '[{"field": "name", "operator": "CONTAIN", "value": "Shop"}]'
- **Character vector shortcut:** String format "<field> <operator> <value>", e.g., c("account_status EQUAL 1", "name CONTAIN Shop").

API supported filterable fields and operators:

- name: Ad account name. Supported operator: CONTAIN. Available on both personal profile (me/adaccounts) and Business Manager.
- account_status: Status code (1 = ACTIVE, 2 = DISABLED, etc.). Supported operator: EQUAL. (Note: operator IN is not supported by Facebook on this endpoint).
- account_id: Account ID without prefix. Supported operator: EQUAL (on me/adaccounts).
- id: Account ID with act_ prefix (e.g. "act_12345"). Supported operator: IN (on Business Manager owned_ad_accounts / client_ad_accounts).

Note: If you need to retrieve details for a specific set or vector of account IDs, prefer using [fbGetAdAccount](#), which queries account nodes directly and does not have edge filtering constraints.

Value

Data frame (tibble) with ad account list and metadata.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
# Load all ad accounts
all_accounts <- fbGetAdAccounts()

# Filter active accounts only (account_status EQUAL 1)
active_accounts <- fbGetAdAccounts(
  source_id = "xxxxxxxxxxxxx",
  filtering = list(
    list(field = "account_status", operator = "EQUAL", value = 1)
  )
)

# Filter accounts by name using JSON string
shop_accounts <- fbGetAdAccounts(
  source_id = "xxxxxxxxxxxxx",
  filtering = '[{"field":"name","operator":"CONTAIN","value":"Shop"}]'
)

# Filter accounts in a Business Manager by ID
bm_accounts <- fbGetAdAccounts(
  source_id = "xxxxxxxxxxxxx",
  filtering = list(
    list(field = "id", operator = "IN", value = list("act_111111", "act_222222"))
  )
)

# Or use fbGetAdAccount() for direct querying of specific IDs
specific_accounts <- fbGetAdAccount(
  account_id = c("1872462700348355", "3281404162010857")
)
```

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

```
fbGetAdAccountsConversions
```

Get list of custom conversion from ad accounts.

Description

Custom conversions on Facebook allows you to optimize and track actions without having to add anything to your Facebook pixel base code. They also allow you to optimize for and track actions that are different from the 9 standard events that come with the Facebook pixel.

Usage

```
fbGetAdAccountsConversions(
  accounts_id = getOption("rfacebookstat.accounts_id"),
  api_version = getOption("rfacebookstat.api_version"),
  username    = getOption("rfacebookstat.username"),
  token_path  = fbTokenPath(),
  access_token = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	Your account ID, for example 'act_1111111111111111'.
access_token	Your facebook API token.
username	your username on Facebook
token_path	path to dir with credentials
api_version	Current Facebook API version.

Value

Data frame with custom conversion list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
conversions <- fbGetAdAccountsConversions()

## End(Not run)
```

fbGetAdAccountUsers *Get User List From Account*

Description

fbGetAdAccountUsers get data frame with adaccounts user list in R.

Usage

```
fbGetAdAccountUsers(
  accounts_id = getOption("rfacebookstat.accounts_id"),
  business_id = getOption("rfacebookstat.business_id"),
  api_version = getOption("rfacebookstat.api_version"),
  console_type = "progressbar",
  username = getOption("rfacebookstat.username"),
  token_path = fbTokenPath(),
  access_token = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	vector with ID of your ad account, for example c("act_00001","act_00002").
business_id	ID by your Business Manager
access_token	Your facebook API token.
console_type	Console output type, "progressbar" or "message"
username	your username on Facebook
token_path	path to dir with credentials
api_version	Current Facebook API version.

Value

Data frame with ad account list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
accountUsers <- fbGetAdAccountUsers(accounts_id = c("act_00001","act_00002"),
                                     bussiness_id = 1111111,
                                     access_token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

## End(Not run)
```

`fbGetAdAccountUsersPermissions`*Get ad account user list with him permissions.*

Description

`fbGetAdAccountUsersPermissions` get ad account user list with him permissions. in R.

Usage

```
fbGetAdAccountUsersPermissions(  
  accounts_id = getOption("rfacebookstat.accounts_id"),  
  api_version = getOption("rfacebookstat.api_version"),  
  console_type = "progressbar",  
  username     = getOption("rfacebookstat.username"),  
  token_path   = fbTokenPath(),  
  access_token = getOption("rfacebookstat.access_token"))
```

Arguments

<code>accounts_id</code>	vector with ID of your ad account, for example <code>c("act_00001","act_00002")</code> .
<code>access_token</code>	Your facebook API token.
<code>console_type</code>	Console output type.
<code>username</code>	your username on Facebook
<code>token_path</code>	path to dir with credentials
<code>api_version</code>	Current Facebook API version.

Value

Data frame with ad account user permissions list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:  
UP <- fbGetAdAccountUsersPermissions(accounts_id = c("act_00001","act_00002"),  
                                     api_version = "v2.10",  
                                     access_token = "xxxxxx")  
  
## End(Not run)
```

fbGetAdCreative

*Get creative list from facebook marketing API***Description**

Get ads creatives list with parameters from facebook marketing API. Format which provides layout and contains content for the ad.

Usage

```
fbGetAdCreative(
  accounts_id = getOption("rfacebookstat.accounts_id"),
  filtering   = NULL,
  api_version = getOption("rfacebookstat.api_version"),
  username    = getOption("rfacebookstat.username"),
  token_path  = fbTokenPath(),
  access_token = getOption("rfacebookstat.access_token"),
  limit       = 50)
```

Arguments

accounts_id	Your account ID, for example 'act_1111111111111111'.
filtering	filter creatives
api_version	Current Facebook API version.
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token.
limit	Number of rows per API call

Value

Data frame with ads list.

1. id	The ID of this creative.
2. name	The name of the creative in the creative library.
3. status	The status of this creative.
4. url_tags	A set of query string parameters which will replace or be appended to urls clicked from page post ads, message of the post, and canvas app install creatives only.
5. account_id	The ID of the ad account that this ad belongs to.
5. page_id	ID of a Facebook page. An unpublished page post will be created on this page. User must have Admin or Editor role for this page.
5. link	Link url.

- 6. message The main body of the post.
- 7. caption Link caption.
- 8. attachment_style The style of the attachment
- 9. description Link description.
- 10. image_hash Hash of an image in your image library with Facebook.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
library(rfacebookstat)
fb_token <- fbGetToken(app_id = 00000000)
ads_creative <- fbGetAdCreative(accounts_id = "act_1111111111111111",
                                api_version = 'v3.0',
                                access_token = fb_token)

## End(Not run)
```

fbGetAds

Get ads list from facebook marketing API

Description

Get ads list with parameters from facebook marketing API. Contains information to display an ad and associate it an ad set. Each ad is associated with an ad set and all ads in a set have the same daily or lifetime budget, schedule, and targeting. Creating multiple ads in an ad set helps optimize their delivery based on variations in images, links, video, text or placements.

Usage

```
fbGetAds(
  accounts_id = getOption("rfacebookstat.accounts_id"),
  fields      = c("id",
                  "name",
                  "object_url",
                  "adlabels",
                  "adset_id",
                  "bid_amount",
                  "bid_type",
                  "campaign_id",
                  "account_id",
                  "configured_status",
```

```

        "effective_status",
        "creative",
        "ad_disclaimer"),
    api_version = getOption("rfacebookstat.api_version"),
    username    = getOption("rfacebookstat.username"),
    token_path  = fbTokenPath(),
    access_token = getOption("rfacebookstat.access_token"),
    limit       = 5000)

```

Arguments

accounts_id	Your account ID, for example 'act_1111111111111111'.
fields	Character vector of fields what you need, list of available fields < https://developers.facebook.com/docs/commerce/marketing-api/reference/adgroup >.
api_version	Current Facebook API version.
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token.
limit	Row limit per 1 request.

Value

Data frame with ads list.

1. id The ID of this ad.
2. name Name of the ad.
3. creative_id The ID or creative spec of the ad creative to be used by this ad.
4. adset_id The ID of the ad set, required on creation.
5. campaign_id ID of the ad campaign that contains this ad.
6. account_id The ID of the ad account that this ad belongs to.
7. bid_amount Bid amount for this ad which will be used in auction instead of the ad set bid_amount, if specified. Any updates to the ad set bid_amount will overwrite this value with the new ad set value.
8. bid_type Bid type, one of CPC, CPM, MULTI_PREMIUM, ABSOLUTE_OCPM, CPA
9. configured_status The configured status of the ad.
10. effective_status The effective status of the ad. The status could be effective either because of its own status, or the status of its parent units.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
library(rfacebookstat)
fb_token <- fbGetToken(app_id = 0000000)
ads <- fbGetAds(accounts_id = "act_1111111111111111",
                 api_version = 'v3.0',
                 access_token = fb_token)

## End(Not run)
```

fbGetAdSets

*Get creative list from facebook marketing API***Description**

Get ads creatives list with parameters from facebook marketing API. An ad set is a group of ads that share the same daily or lifetime budget, schedule, bid type, bid info, and targeting data. Ad sets enable you to group ads according to your criteria, and you can retrieve the ad-related statistics that apply to a set.

Usage

```
fbGetAdSets(accounts_id = getOption("rfacebookstat.accounts_id"),
             api_version = getOption("rfacebookstat.api_version"),
             username     = getOption("rfacebookstat.username"),
             token_path   = fbTokenPath(),
             access_token = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	Your account ID, for example 'act_1111111111111111'.
api_version	Current Facebook API version.
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token.

Value

Data frame with ads list.

1. id Ad set ID.
2. name Name of ad set.
3. account_id Ad Account ID.
4. budget_remaining Remaining budget.

5. `configured_status`

The status set at the ad set level. It can be different from the effective status due to its parent campaign. Prefer using 'status' instead of this, one of ACTIVE, PAUSED, DELETED, ARCHIVED.

6. `effective_status`

The effective status of the ad set, which can be either its own status or caused by its parent campaign, one of ACTIVE, PAUSED, DELETED, PENDING_REVIEW, DISAPPROVED, PREAPPROVED, PENDING_BILLING_INFO, CAMPAIGN_PAUSED, ARCHIVED, ADSET_PAUSED.

7. `status`

The status set at the ad set level. It can be different from the effective status due to its parent campaign. The field returns the same value as 'configured_status', and is the suggested one to use (ACTIVE, PAUSED, DELETED, ARCHIVED).

8. `created_time`

Created time.

9. `bid_strategy`

Choose bid strategy for this ad set to suit your specific business goals. Each strategy has tradeoffs and may be available for certain optimization_goals: **LOWEST_COST_WITHOUT_CAP**: Designed to get the most results for your budget based on your ad set optimization_goal without limiting your bid amount. This is the best strategy if you care most about cost efficiency. However with this strategy it may be harder to get stable average costs as you spend. This strategy is also known as automatic bidding. Learn more in Ads Help Center, About bid strategies: Lowest cost. **LOWEST_COST_WITH_BID_CAP**: Designed to get the most results for your budget based on your ad set optimization_goal while limiting actual bid to your specified amount. With a bid cap you have more control over your cost per actual optimization event. However if you set a limit which is too low you may get less ads delivery. If you select this, you must provide a bid cap with the bid_amount field. Note: during creation this bid strategy is set if you provide bid_amount only. This strategy is also known as manual maximum-cost bidding. Learn more in Ads Help Center, About bid strategies: Lowest cost. **TARGET_COST**: Maintains a stable average cost based on your optimization_goal as you raise your ad set budget. Select this strategy if you care most about maintaining stable average costs for your ads. However, note that this strategy may be less cost-efficient than the lowest cost strategy. If you select this, you must provide a target cost with the bid_amount field. Target cost bid strategy is also known as manual average-cost bidding. Learn more in Ads Help Center, About bid strategies: Target cost. Note: If you enable campaign budget optimization, you should set bid_strategy at the parent campaign level.

10. `pacing_type`

Defines the pacing type, standard by default or using ad scheduling

Author(s)

Alexey Seleznev

Examples

Not run:

```
library(rfacebookstat)
fb_token <- fbGetToken(app_id = 00000000)
adsets <- fbGetAdSets(accounts_id = "act_1111111111111111",
                      api_version = 'v3.0',
                      access_token = fb_token)

## End(Not run)
```

fbGetAdVideos

Get videos list from ad accounts

Description

Load all videos from facebook ad accounts.

Usage

```
fbGetAdVideos(accounts_id = getOption("rfacebookstat.accounts_id"),
              api_version = getOption("rfacebookstat.api_version"),
              username     = getOption("rfacebookstat.username"),
              token_path   = fbTokenPath(),
              access_token = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	Your account ID, for example 'act_1111111111111111'.
api_version	Current Facebook API version.
username	Your username on Facebook
token_path	Path to dir with credentials
access_token	Your facebook API token.

Value

Data frame with video parameters list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
ads <- fbGetAdVideos()

## End(Not run)
```

fbGetApps*Get all the apps under a project*

Description

Get data frame with list apps under a project.

Usage

```
fbGetApps(accounts_id = getOption("rfacebookstat.accounts_id"),
  api_version = getOption("rfacebookstat.api_version"),
  username = getOption("rfacebookstat.username"),
  token_path = fbTokenPath(),
  access_token = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	Facebook Ad Account ID.
api_version	Current Facebook API version.
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token.

Value

Data frame with apps list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
accounts <- fbGetAccounts()
fbApps <- fbGetApps(accounts$id)

## End(Not run)
```

fbGetBusinessInvoices *Get Business Invoices*

Description

fbGetBusinessInvoices get invoices for a Facebook Business Manager.

Usage

```
fbGetBusinessInvoices(  
  business_id,  
  start_date = NULL,  
  end_date = NULL,  
  issue_start_date = NULL,  
  issue_end_date = NULL,  
  invoice_id = NULL,  
  type = NULL,  
  api_version = getOption("rfacebookstat.api_version"),  
  username = getOption("rfacebookstat.username"),  
  token_path = fbTokenPath(),  
  access_token = getOption("rfacebookstat.access_token")  
)
```

Arguments

business_id	ID of the Facebook Business Manager.
start_date	Start date for the billing period (YYYY-MM-DD).
end_date	End date for the billing period (YYYY-MM-DD).
issue_start_date	Start date when the invoice was issued (YYYY-MM-DD).
issue_end_date	End date when the invoice was issued (YYYY-MM-DD).
invoice_id	Filter by a specific invoice ID.
type	Type of the document, e.g., "INV", "PRO_FORMA", "CM", "DM".
api_version	Current Facebook API version.
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token

Value

Data frame with business invoices.

Author(s)

Alexey Seleznev

fbGetBusinessManagers *Get avable business Managers.*

Description

fbGetBusinessManagers get data frame with avable bussines managers list in R.

Usage

```
fbGetBusinessManagers(api_version = getOption("rfacebookstat.api_version"),
                      username     = getOption("rfacebookstat.username"),
                      token_path   = fbTokenPath(),
                      access_token = getOption("rfacebookstat.access_token"))
```

Arguments

access_token	Your facebook API token.
username	your username on Facebook
token_path	path to dir with credentials
api_version	Current Facebook API version.

Value

Data frame with ad account list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
BM <- fbGetBusinessManagers(api_version = "v2.10", access_token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

## End(Not run)
```

`fbGetBusinessManagersUsers`*Get a list of business manager users*

Description

Loading user list from business managers.

Usage

```
fbGetBusinessManagersUsers(  
  business_ids = getOption("rfacebookstat.business_id"),  
  user_types   = c('business_users',  
                   'system_users',  
                   'pending_users'),  
  api_version  = getOption("rfacebookstat.api_version"),  
  username     = getOption("rfacebookstat.username"),  
  token_path   = fbTokenPath(),  
  access_token = getOption("rfacebookstat.access_token"))
```

Arguments

<code>business_ids</code>	IDs by your Business Manager
<code>user_types</code>	user types: <code>business_users</code> , <code>system_users</code> , <code>pending_users</code> .
<code>username</code>	your username on Facebook
<code>token_path</code>	path to dir with credentials
<code>access_token</code>	Your facebook API token.
<code>api_version</code>	Current Facebook API version.

Value

Data frame with bussiness managers user list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:  
options(rfacebookstat.business_id = 000000000000)  
  
bm_users <- fbGetBusinessManagersUsers()  
  
## End(Not run)
```

fbGetBusinessUserAdAccounts

Get a list of accounts for a business manager user

Description

Loading account list for business manager user.

Usage

```
fbGetBusinessUserAdAccounts(
  business_users_id = NULL,
  business_id       = getOption("rfacebookstat.business_id"),
  api_version       = getOption("rfacebookstat.api_version"),
  username          = getOption("rfacebookstat.username"),
  token_path        = fbTokenPath(),
  access_token      = getOption("rfacebookstat.access_token"))
```

Arguments

business_users_id	ID by business manager user.
business_id	ID by your Business Manager
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token.
api_version	Current Facebook API version.

Value

Data frame with ad account list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
options(rfacebookstat.business_id = 000000000000,
  rfacebookstat.access_token = "abcde12345...")

my_accs <- fbGetBusinessUserAdAccounts(
  business_users_id = 777423445991332
)

## End(Not run)
```

fbGetCampaigns*Get campaign list from facebook marketing API*

Description

Get campaign list with parameters from facebook marketing API.

Usage

```
fbGetCampaigns(accounts_id = getOption("rfacebookstat.accounts_id"),
               api_version = getOption("rfacebookstat.api_version"),
               username     = getOption("rfacebookstat.username"),
               token_path   = fbTokenPath(),
               access_token = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	Your account ID, for example 'act_1111111111111111'.
api_version	Current Facebook API version.
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token.

Value

Data frame with campaign list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
library(rfacebookstat)
fb_token <- fbGetToken(app_id = 00000000)
camp <- fbGetCampaigns(accounts_id = "act_1111111111111111",
                      api_version = 'v3.0',
                      access_token = fb_token)

## End(Not run)
```

`fbGetCatalogs`*Get catalogs*

Description

Load catalogs from business maneger.

Usage

```
fbGetCatalogs(business_id = getOption("rfacebookstat.business_id") ,
              api_version = getOption("rfacebookstat.api_version"),
              username     = getOption("rfacebookstat.username"),
              token_path   = fbTokenPath(),
              access_token = getOption("rfacebookstat.access_token"))
```

Arguments

<code>business_id</code>	ID by your Business Manager
<code>api_version</code>	Current Facebook API version.
<code>username</code>	your username on Facebook
<code>token_path</code>	path to dir with credentials
<code>access_token</code>	Your API facebook token

Value

Data frame with project list

Author(s)

Alexey Seleznev

Examples

```
## Not run:
bc <- fbGetCatalogs(business_id = 1111111111,
                    access_token = "xxxxxxx")

## End(Not run)
```

fbGetCostData	<i>Get facebook ads cost data.</i>
---------------	------------------------------------

Description

Get cost data for loading it into Google Analytics

Usage

```
fbGetCostData(  
  accounts_id = getOption("rfacebookstat.accounts_id"),  
  date_start   = Sys.Date() - 30,  
  date_stop    = Sys.Date(),  
  utm_source   = "facebook",  
  utm_medium   = "cpc",  
  username     = getOption("rfacebookstat.username"),  
  token_path   = fbTokenPath(),  
  access_token = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	ID of your ad account.
date_start	Start reporting date.
date_stop	End reporting day
utm_source	UTM source name for Google Analytics.
utm_medium	UTM medium name for Google Analytics.
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token

Details

Before loading data create data source at Google Analytics.

Value

Data frame with statistic.

Author(s)

Alexey Seleznev

Examples

```
## Not run:

# Use googleAnalyticsR for upload data into Google Analytics
library(googleAnalyticsR)

# load cost data
cost_data <- fbGetCostData(accounts_id = "act_11111",
                           date_start  = "2020-05-10",
                           date_stop   = "2020-05-14",
                           username    = "fb_login")

# upload into GA source
ga_custom_upload_file(accountId      = xxxx,
                      webPropertyId  = "UA-xxxx-1",
                      customDataSourceId = 'abcdefg',
                      cost_data)

## End(Not run)
```

fbGetLogins

*Get all authorization logins***Description**

Get all your authorized facebook logins

Usage

```
fbGetLogins(
  token_path = fbTokenPath(),
  set_login  = TRUE)
```

Arguments

token_path	path to dir with credentials
set_login	if TRUE you can choose and change current session login

Author(s)

Alexey Seleznev

Examples

```
## Not run:
fbGetLogins()

## End(Not run)
```

fbGetLongTimeToken	<i>Get API facebook long time token.</i>
--------------------	--

Description

Change your short time token to long time token.

Usage

```
fbGetLongTimeToken(client_id= NULL,client_secret = NULL,fb_exchange_token = NULL)
```

Arguments

client_id	Your App ID.
client_secret	Your App secret.
fb_exchange_token	Your short time API token.

Value

API token

Author(s)

Alexey Seleznev

fbGetMarketingStat	<i>Get statistic by ad accounts.</i>
--------------------	--------------------------------------

Description

fbGetMarketingStat is main function of rfacebookstat package, intended for load statistic data by your ad, adset, campaign or account

Usage

```
fbGetMarketingStat(  
  accounts_id = getOption("rfacebookstat.accounts_id"),  
  sorting = NULL, level = "account", breakdowns = NULL,  
  action_breakdowns = NULL,  
  fields = "account_id,campaign_name,impressions,clicks,reach,spend",  
  filtering = NULL, date_start = NULL,  
  date_stop = NULL, date_preset = "last_30d",  
  attribution_window = NULL,  
  api_version = getOption("rfacebookstat.api_version"),
```

```

action_report_time = NULL, interval = "day",
use_unified_attribution_setting = FALSE,
use_account_attribution_setting = FALSE,
console_type = "progressbar", request_speed = "normal",
fetch_by = NULL,
  username = getOption("rfacebookstat.username"),
  token_path = fbTokenPath(),
access_token = getOption("rfacebookstat.access_token"))

```

Arguments

accounts_id	ID of your ad account.
sorting	Field to sort the result, and direction of sorting. You can specify sorting direction by appending "_ascending" or "_descending" to the sort field. For example, "reach_descending". This array supports no more than one element. By default, the sorting direction is ascending.
level	Represents the level of result. Available ad, adset, campaign, account.
action_breakdowns	group results in the actions field. You can use the following breakdowns for action_breakdowns, for details go , or see details
breakdowns	Permutations marked with an asterisk (*) can be joined with action_type and action_target_id. Available age, country, gender, frequency_value, hourly_stats_aggregated_by_advertiser_time_zone, hourly_stats_aggregated_by_audience_time_zone, impression_device, place_page_id, placement, device_platform, product_id, region. See details.
fields	List of fields which you want get in R.
filtering	Vector of filtering or JSON string with array of filtering parameters, on example "ad.effective_status IN ARCHIVED" [{"field":"'ad.effective_status/'", "operator":"'&in'"}] see filtering block for more examples
date_start	Start reporting date.
date_stop	End reporting day
date_preset	Represents a relative time range. This field is ignored if time_range or time_ranges is specified. One of: today, yesterday, this_month, last_month, this_quarter, lifetime, last_3d, last_7d, last_14d, last_28d, last_30d, last_90d, last_week_mon_sun, last_week_sun_sat, last_quarter, last_year, this_week_mon_today, this_week_sun_today, this_year
attribution_window	The conversion attribution window provides timeframes that define when we attribute an event to an ad on Facebook. See Attribution Window sections.
request_speed	Speed between API request, "normal", "fast" or "slow", depend of you API access level.
fetch_by	Character, split your request by time interval, one of: day, week, month, quarter, year.
api_version	Current Facebook API version.
action_report_time	Determines the report time of action stats.

interval	Character value for split by time interval, one of "day", "week", "month", "quarter", "year". By default = "day"
use_unified_attribution_setting	When this parameter is set to true, your ads results will be shown using unified attribution settings defined at ad set level and parameter use_account_attribution_setting will be ignored.
use_account_attribution_setting	When this parameter is set to true, your ads results will be shown using the attribution settings defined for the ad account.
console_type	Character value for manage console output message, one of "progressbar", "message". By default = "progressbar". See more at Detail
username	your username on Facebook
token_path	path to dir with credentials
access_token	Your facebook API token

Details

Console_type parameters, if chose "progressbar" you can see load progress in percent, and if chose "message" you get message about loading process.

Value

Data frame with statistic.

Breakdowns

Allowed values for breakdowns.

- ad_format_asset
- age
- body_asset
- call_to_action_asset
- country
- description_asset
- gender
- image_asset
- impression_device
- link_url_asset
- product_id
- region
- title_asset
- video_asset
- dma

- frequency_value
- hourly_stats_aggregated_by_advertiser_time_zone
- hourly_stats_aggregated_by_audience_time_zone
- place_page_id
- publisher_platform
- platform_position
- device_platform

Action Breakdowns

Group results in the actions field. You can use the following breakdowns for action_breakdowns. Now you can use next action breakdowns:

action_device The device on which the conversion event you're tracking occurred. For example, "Desktop" if someone converted on a desktop computer.

action_destination The destination where people go after clicking on your ad. This could be your Facebook Page, an external URL for your conversion pixel or an app configured with the software development kit (SDK).

action_reaction The number of reactions on your ads or boosted posts. The reactions button on an ad allows people to share different reactions on its content: Like, Love, Haha, Wow, Sad or Angry.

action_target_id The id of destination where people go after clicking on your ad. This could be your Facebook Page, an external URL for your conversion pixel or an app configured with the software development kit (SDK).

action_type The kind of actions taken on your ad, Page, app or event after your ad was served to someone, even if they didn't click on it. Action types include Page likes, app installs, conversions, event responses and more.

action_type,action_reaction Together of action types and reactions breakdown.

Available Combinations Of Breakdowns

Grouping types marked with an asterisk (*) can be combined with action_type, action_target_id, and action_destination (action_target_id).

- action_type *
- action_target_id *
- action_device *
- action_device, impression_device *
- action_device, publisher_platform *
- action_device, publisher_platform, impression_device *
- action_device, publisher_platform, platform_position *
- action_device, publisher_platform, platform_position, impression_device *
- action_reaction

- action_type, action_reaction
- age *
- gender *
- age, gender *
- country *
- region *
- publisher_platform *
- publisher_platform, impression_device *
- publisher_platform, platform_position *
- publisher_platform, platform_position, impression_device *
- product_id *

Attribution Window

The conversion attribution window provides time intervals that determine the attribution period of an event for advertising on Facebook. For background information, see Facebook Ads Help Center, [How Attribution Reporting Works](#). We measure the actions that occur when a conversion event occurs and look back in time 1-day, 7-days, and 28 days. To view actions attributed to different attribution windows, use `attribution_window`.

account_default Use the account level attribution window setting

default The FB default attribution window is 1 day views, 28 day clicks

inline Inline attribution only (0 day views, 0 day clicks)

1d_view 1 day views, 0 day clicks

7d_view 7 day views, 0 day clicks

28d_view 28 day views, 0 day clicks

1d_click 0 day views, 1 day clicks

7d_click 0 day views, 7 day clicks

28d_click 0 day views, 28 day clicks

1d_view_1d_click 1 day views, 1 day clicks

7d_view_1d_click 7 day views, 1 day clicks

28d_view_1d_click 28 day views, 1 day clicks

1d_view_7d_click 1 day views, 7 day clicks

1d_view_28d_click 1 day views, 28 day clicks

28d_view_28d_click 28 day views, 28 day clicks

Also you can get more than one attribution window in one request, for example `attribution_window = c('default', '1d_view', '28d_view', '28d_click')`

Filtering

Filters on the report data. This parameter is an array of filter objects. You can set string vector or JSON string with field, operator and value.

field Field for filtering.

operator One of EQUAL, NOT_EQUAL, GREATER_THAN, GREATER_THAN_OR_EQUAL, LESS_THAN, LESS_THAN_OR_EQUAL, IN_RANGE, NOT_IN_RANGE, CONTAIN, NOT_CONTAIN, IN, NOT_IN, STARTS_WITH, ANY, ALL, AFTER, BEFORE, NONE.

value Field value for filtering.

Simple filtering:

Example vector `filtering = "publisher_platform IN instagram"`

Example JSON `filtering = "[{'field': 'publisher_platform', 'operator': 'IN', 'value': ['instagram']}]"`

Example of using two or more conditions: `filtering = "[{'field': 'clicks', 'operator': 'LESS_THAN', 'value': '500'}, {'field': 'impressions', 'operator': 'GREATER_THAN', 'value': '20000'}]"`

Filtering by two or more conditions:

Example vector `filtering = c("clicks LESS_THAN 500", "impressions GREATER_THAN 20000")`

Example JSON `filtering = '[{"field": "clicks", "operator": "LESS_THAN", "value": "500"}, {"field": "impressions", "operator": "GREATER_THAN", "value": "20000"}]'`

Filtering with operators IN_RANGE, NOT_IN_RANGE, IN, NOT_IN

Example vector `filtering = 'publisher_platform IN instagram, facebook'`

Example JSON `filtering = '[{"field": "publisher_platform", "operator": "IN", "value": ["instagram", "facebook"]}]'`

Author(s)

Alexey Seleznev

See Also

All available field, and description of all parameters you can see on this [link](#). More information about breakdowns and action breakdowns on this [link](#). More information about action field in this [documentation](#).

Examples

```
## Not run:
fbStat <- fbGetMarketingStat(accounts_id = "act_XXXXXXXXXXXX",
                             level = "campaign",
                             fields = "account_name,campaign_name,impressions",
                             breakdowns = "device_platform",
                             date_start = "2016-08-01",
                             date_stop = "2016-08-10",
                             interval = "day",
                             access_token = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX")

## End(Not run)
```

`fbGetPages`*Get pages list*

Description

Load data about all avable pages

Usage

```
fbGetPages(accounts_id = getOption("rfacebookstat.accounts_id"),
            api_version = getOption("rfacebookstat.api_version"),
            username    = getOption("rfacebookstat.username"),
            token_path  = fbTokenPath(),
            access_token = getOption("rfacebookstat.access_token"))
```

Arguments

<code>accounts_id</code>	Facebook Ad Account ID.
<code>api_version</code>	Current Facebook API version.
<code>username</code>	your username on Facebook
<code>token_path</code>	path to dir with credentials
<code>access_token</code>	Your facebook API token.

Value

Data frame with pages list

Author(s)

Alexey Seleznev

Examples

```
## Not run:
accounts <- fbGetAccounts()
fbPages <- fbGetPages(accounts$id)

## End(Not run)
```

fbGetSettings	<i>Get settings.</i>
---------------	----------------------

Description

Get settings of rfacebookstat package.

Usage

```
fbGetSettings()
```

Author(s)

Alexey Seleznev

Examples

```
## Not run:  
fbGetSettings()  
  
## End(Not run)
```

fbGetToken	<i>Get API facebook token.</i>
------------	--------------------------------

Description

Get API facebook token for access to facebook ads API.

Usage

```
fbGetToken(  
  app_id = NULL,  
  scopes = c("ads_read", "business_management", "pages_manage_ads", "ads_management",  
            "public_profile")  
)
```

Arguments

app_id	ID of your Facebook App
scopes	Permissions provide a way for your app to access data from Facebook. For detail see documentation

Value

API token

Examples

```
## Not run:
tkn <- fbGetToken()

## End(Not run)
```

fbGetUserAdAccounts	<i>User's ad account list.</i>
---------------------	--------------------------------

Description

The advertising accounts to which this person has access.

Usage

```
fbGetUserAdAccounts(user_id      = "me",
                    api_version  = getOption("rfacebookstat.api_version"),
                    username     = getOption("rfacebookstat.username"),
                    token_path   = fbTokenPath(),
                    access_token = getOption("rfacebookstat.access_token"))
```

Arguments

user_id	Facebook user id.
access_token	Your facebook API token.
username	your username on Facebook
token_path	path to dir with credentials
api_version	Current Facebook API version.

Value

Data frame with ad account list.

Author(s)

Alexey Seleznev

Examples

```
## Not run:
fbAccounts <- fbGetUserAdAccounts()

## End(Not run)
```

fbSetters	<i>Set rfacebookstat options</i>
-----------	----------------------------------

Description

Set rfacebookstat options in current R session

Usage

```
fbSetUsername(username)
fbSetAccount(accounts_ids)
fbSetBusinessId(business_ids)
fbSetTokenPath(token_path)
fbSetApiVersion(api_version)
```

Arguments

accounts_ids	Facebook Ad Account ID.
business_ids	IDs by your Business Manager
username	your username on Facebook
token_path	path to dir with credentials
api_version	Current Facebook API version.

Author(s)

Alexey Seleznev

fbUpdateAdAccountUsers	<i>Add users and update permission list.</i>
------------------------	--

Description

fbUpdateAdAccountUsers add users and update permission list for users in ad accounts at facebook.

Usage

```
fbUpdateAdAccountUsers(
  user_ids      = NULL,
  role          = "advertiser",
  accounts_id   = getOption("rfacebookstat.accounts_id"),
  api_version   = getOption("rfacebookstat.api_version"),
  username      = getOption("rfacebookstat.username"),
  token_path    = fbTokenPath(),
  access_token  = getOption("rfacebookstat.access_token"))
```

Arguments

accounts_id	vector with ID of your ad account, for example c("act_00001","act_00002").
role	Character or integer value, one of administrator, advertiser, ad manager, analyst, sales, direct sales, 1001, 1002, 1003,1004 (https://developers.facebook.com/documentation/ads-commerce/marketing-api/aduser/v2.12).
access_token	Your facebook API token.
user_ids	vector with users ids who need remove from ad account.
username	your username on Facebook
token_path	path to dir with credentials
api_version	Current Facebook API version.

Details

Requires: ads_management permission is required. You need 'Administrator' access to the ad account to remove users

Roles: 1001 = Administrator access 1002 = Advertiser, or ad manager, access 1003 = Analyst access 1004 = Direct sales access. For limited, directly managed accounts.

Permissions: 1: ACCOUNT_ADMIN: modify list of people associated with the account. 2: AD-MANAGER_READ: view campaigns and ads 3: ADMANAGER_WRITE: manage campaigns and ads 4: BILLING_READ: view account billing information 5: BILLING_WRITE: modify account billing information 7: REPORTS: run reports 9, 10 - Apply to some directly managed accounts. Not currently needed for Marketing API calls

For more details go link (<https://developers.facebook.com/documentation/ads-commerce/marketing-api/aduser/>)

Author(s)

Alexey Seleznev

Examples

```
## Not run:
#Attach package
library(rfacebookstat)

#Get token, where 0000000000000000 is id of your app in facebook
tk <- fbGetToken(0000000000000000)

#Add user with id 001 and 002 to accounts act_0001 and act_0002
fbUpdateAdAccountUsers(user_ids = c("001","002"),
  accounts_id = c("act_0001","act_0002"),
  role = "advertiser",
  api_version = "v2.12",
  access_token = tk)

## End(Not run)
```

Index

*** package**
 rfacebookstat-package, [2](#)

fbAuth, [3](#)
fbDeleteAdAccountUsers, [4](#)
fbDownloadBusinessInvoices, [6](#)
fbGetAdAccount, [7](#), [10](#)
fbGetAdAccountCustomAudiences, [8](#)
fbGetAdAccounts, [9](#)
fbGetAdAccountsConversions, [11](#)
fbGetAdAccountUsers, [12](#)
fbGetAdAccountUsersPermissions, [13](#)
fbGetAdCreative, [14](#)
fbGetAds, [15](#)
fbGetAdSets, [17](#)
fbGetAdVideos, [19](#)
fbGetApps, [20](#)
fbGetBusinessInvoices, [21](#)
fbGetBusinessManagers, [22](#)
fbGetBusinessManagersUsers, [23](#)
fbGetBusinessUserAdAccounts, [24](#)
fbGetCampaigns, [25](#)
fbGetCatalogs, [26](#)
fbGetCostData, [27](#)
fbGetLogins, [28](#)
fbGetLongTimeToken, [29](#)
fbGetMarketingStat, [29](#)
fbGetPages, [35](#)
fbGetSettings, [36](#)
fbGetToken, [36](#)
fbGetUserAdAccounts, [37](#)
fbSetAccount (fbSetters), [38](#)
fbSetApiVersion (fbSetters), [38](#)
fbSetBusinessId (fbSetters), [38](#)
fbSetters, [38](#)
fbSetTokenPath (fbSetters), [38](#)
fbSetUsername (fbSetters), [38](#)
fbUpdateAdAccountUsers, [38](#)

rfacebookstat (rfacebookstat-package), [2](#)