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/docs/marketing-apis/>.

Details

The DESCRIPTION file:

Package: rfacebookstat
Type: Package
Title: Load Data from Facebook API Marketing
Version: 2.0.3
Date: 2020-05-06
Author: Alexey Seleznev <selesnow@gmail.com>
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/docs/marketing-apis/>.
Depends: R (>= 3.5.0)
BugReports: https://github.com/selesnow/rfacebookstat/issues
Imports: httr, jsonlite, dplyr, purrrr, tidyr (>= 1.0.0), stringr, tidyselect
URL: http://selesnow.github.io/rfacebookstat
License: GPL-2
Index of help topics:

- fbAuth: Authorization in Facebook.
- fbDeleteAdAccountUsers: Delete users from ad accounts.
- fbGetAdAccountUsers: Get User List From Account
- fbGetAdAccountUsersPermissions: Get ad account user list with him permissions.
- fbGetAdAccounts: Get avable ad account list.
- fbGetAdAccountsConversions: Get list of custom conversion from ad accounts.
- fbGetAdCreative: Get creative list from facebook marketing API
- fbGetAdSets: Get creative list from facebook marketing API
- fbGetAdVideos: Get videos list from ad accounts
- fbGetAds: Get ads list from facebook marketing API
- fbGetApps: Get all the apps under a project
- fbGetBusinessUserAdAccounts: Get a list of accounts for a business manager user
- fbGetCampaigns: Get campaign list from facebook marketing API
- fbGetCatalogs: Get catalogs
- fbGetLongTimeToken: Get API facebook long time token.
- fbGetMarketingStat: Get statistic by ad accounts.
- fbGetPages: Get pages list
- fbGetSettings: Get settings.
- fbGetToken: Get API facebook token.
- fbGetUserAdAccounts: User's ad account list.
- fbUpdateAdAccountUsers: Add users and update permission list.
- rfacebookstat-package: Load Data from Facebook API Marketing

**Author(s)**

Alexey Seleznev <selesnow@gmail.com>

Maintainer: Alexey Seleznev <selesnow@gmail.com>

**See Also**

Examples

## Not run:
library(rfacebookstat)
# Auth
token <- fbGetToken(app_id = 000000000000)

# 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**

```r
fbAuth(app_id = getOption("rfacebookstat.app_id"),
       app_secret = getOption("rfacebookstat.app_secret"),
       username = getOption("rfacebookstat.username"),
       token_path = fbTokenPath(),
       reauth = FALSE,
       skip_option = FALSE)
```

**Arguments**

- `app_id`: ID of Facebook App
- `app_secret`: Secret of Facebook App
- `username`: your username on Facebook
- `token_path`: path to dir with credentials
- `reauth`: boolean, do reauthorization
- `skip_option`: logical, skip access token from options or not

**Value**

API token object

**Author(s)**

Alexey Seleznev
Examples

```r
## Not run:
fbAuth()

## End(Not run)
```

fbDeleteAdAccountUsers

Delete users from ad accounts.

Description

fbDeleteAdAccountUsers is method for remove users access from ad accounts.

Usage

```r
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

```r
## Not run:
#Attach package
tk <- fbAuth(0000000000000)
#Get token, where 0000000000000 is id of your app in facebook
#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)

---

**fbGetAdAccounts**

*Get available ad account list.*

**Description**

`fbGetAdAccounts` get data frame with available in your business manager ad account list in R.

**Usage**

```r
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"))
```

**Arguments**

- `source_id`: Your business manager or project 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**

```r
## Not run:
fbAccounts <- fbGetAdAccounts(source_id = "xxxxxxxxxxxxx",
api_version = "v2.10",
access_token = "xxxxxxxx")

## 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**

```r
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_1111111111111111111`.  
- `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

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

## End(Not run)
```

### Description

`fbGetAdAccountUsers` get data frame with ad accounts user list in R.

### Usage

```r
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
fbGetAdAccountUsersPermissions

Examples

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

## 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

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

Value

Data frame with ad account user permissions list.

Author(s)

Alexey Seleznev
## 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

```r
fbGetAdCreative(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` : 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.
6. `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.
Author(s)

Alexey Seleznev

Examples

```r
## Not run:
library(rfacebookstat)
fb_token <- fbGetToken(app_id = 0000000)
ads_creative <- fbGetAdCreative(accounts_id = "act_11111111111111111",
                        api_version = 'v3.0',
                        access_token = fb_token)
## End(Not run)
```

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

```r
fbGetAds(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_11111111111111111'.
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  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

```r
## Not run:
library(rfacebookstat)
fb_token <- fbGetToken(app_id = 000000)
ads <- fbGetAds(accounts_id = "act_11111111111111111",
                 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

```r
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_11111111111111111'.
- `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

```r
## Not run:
library(rfacebookstat)
fb_token <- fbGetToken(app_id = 000000)
adsets <- fbGetAdSets(accounts_id = "act_11111111111111111",
                        api_version = 'v3.0',
                        access_token = fb_token)

## End(Not run)
```
fbGetAdVideos

Get videos list from ad accounts

Description

Load all videos from faceboob ad accounts.

Usage

```r
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

```r
## 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

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

## End(Not run)
```
fbGetBusinessManagers  

*Get avable business Managers.*

**Description**

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

**Usage**

```r
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**

```r
## Not run:
BM <- fbGetBusinessManagers(api_version = "v2.10", access_token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
## End(Not run)
```
Get a list of accounts for a business manager user

Description

Loading account list for business manager user.

Usage

```r
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`<br>ID by business manager user.
- `business_id`<br>ID by your Business Manager
- `username`<br>your username on Facebook
- `token_path`<br>path to dir with credentials
- `access_token`<br>Your facebook API token.
- `api_version`<br>Current Facebook API version.

Value

Data frame with ad account list.

Author(s)

Alexey Seleznev

Examples

```r
## 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

```r
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_11111111111111111`.
- **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

```r
## Not run:
library(rfacebookstat)
fb_token <- fbGetToken(app_id = 0000000)
camp <- fbGetCampaigns(accounts_id = "act_11111111111111111",
                        api_version = 'v3.0',
                        access_token = fb_token)

## End(Not run)
```
**fbGetCatalogs**  

**Get catalogs**

**Description**

Load catalogs from business manager.

**Usage**

```r
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**

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

**Value**

Data frame with project list

**Author(s)**

Alexey Seleznev

**Examples**

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

## End(Not run)
```
fbGetLongTimeToken  

Get API facebook long time token.

Description

Change your short time token to long time token.

Usage

```r
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  

Get statistic by ad accounts.

Description

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

Usage

```r
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 = Sys.Date() - 30,
    date_stop = Sys.Date(), attribution_window = NULL,
    api_version = getOption("rfacebookstat.api_version"),
    action_report_time = NULL, interval = "day",
```
console_type = "progressbar", request_speed = "normal",
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. Avable 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. Avable 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. You can find list of available fields at Facebook API documentation by this link.
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",/value:[ARCHIVED]}], see filtering block for more examples
date_start  Start reporting date.
date_stop  End reporting day
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.
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"
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'

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**

```r
## Not run:
fbStat <- fbGetMarketingStat(accounts_id = "act_xxxxxxxxxxxxxxx", 
                            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 = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

## End(Not run)
```

---

**fbGetPages**

*Get pages list*

**Description**

Load data about all available pages

**Usage**

```r
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"))
```
**fbGetSettings**

*Get settings.*

**Description**

Get settings of rfacebookstat package.

**Usage**

```r
fbGetSettings()
```

**Author(s)**

Alexey Seleznev

**Examples**

```r
## Not run:
fbGetSettings()
## End(Not run)
```
fbGetToken  
Get API facebook token.

Description
Get API facebook token for access to facebook ads API.

Usage
```r
fbGetToken(app_id = NULL)
```

Arguments

app_id  ID of your Facebook App

Value
API token

Author(s)
Alexey Seleznev

Examples
```r
## Not run:
tkn <- fbGetToken()
## End(Not run)
```

fbGetUserAdAccounts  User’s ad account list.

Description
The advertising accounts to which this person has access.

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

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

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

## End(Not run)
```

Description

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

Usage

```r
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 administratator, advertiser, ad manager, analyst,
sales, direct sales, 1001, 1002, 1003,1004 (https://developers.facebook.com/docs/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/docs/marketing-api/aduser/)

Author(s)

Alexey Seleznev

Examples

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

#get token, where 0000000000000 is id of your app in facebook
tk <- fbGetToken(0000000000000)

#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

*Topic package
  rfacebookstat-package, 2

fbAuth, 4
fbDeleteAdAccountUsers, 5
fbGetAdAccounts, 6
fbGetAdAccountsConversions, 7
fbGetAdAccountUsers, 8
fbGetAdAccountUsersPermissions, 9
fbGetAdCreative, 10
fbGetAds, 11
fbGetAdSets, 13
fbGetAdVideos, 15
fbGetApps, 16
fbGetBusinessManagers, 17
fbGetBusinessUserAdAccounts, 18
fbGetCampaigns, 19
fbGetCatalogs, 20
fbGetLongTimeToken, 21
fbGetMarketingStat, 21
fbGetPages, 26
fbGetSettings, 27
fbGetToken, 28
fbGetUserAdAccounts, 28
fbUpdateAdAccountUsers, 29

rfacebookstat (rfacebookstat-package), 2
rfacebookstat-package, 2