

# Package ‘rfacebookstat’

May 9, 2026

**Type** Package

**Title** Load Data from Facebook API Marketing

**Version** 2.14.0

**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/docs/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](https://www.youtube.com/playlist?list=PLD2LDq8edf4pIt0b-vZTG5AXZK2niJ8_R)

**License** MIT + file LICENSE

**Suggests** knitr, rmarkdown

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**NeedsCompilation** no

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

**Repository** CRAN

**Date/Publication** 2026-04-17 12:40:02 UTC

## Contents

rfacebookstat-package . . . . .	2
fbAuth . . . . .	4
fbDeleteAdAccountUsers . . . . .	5

fbGetAdAccountCustomAudiences . . . . .	6
fbGetAdAccounts . . . . .	7
fbGetAdAccountsConversions . . . . .	8
fbGetAdAccountUsers . . . . .	9
fbGetAdAccountUsersPermissions . . . . .	10
fbGetAdCreative . . . . .	11
fbGetAds . . . . .	13
fbGetAdSets . . . . .	14
fbGetAdVideos . . . . .	16
fbGetApps . . . . .	17
fbGetBusinessManagers . . . . .	18
fbGetBusinessManagersUsers . . . . .	19
fbGetBusinessUserAdAccounts . . . . .	20
fbGetCampaigns . . . . .	21
fbGetCatalogs . . . . .	22
fbGetCostData . . . . .	23
fbGetLogins . . . . .	24
fbGetLongTimeToken . . . . .	25
fbGetMarketingStat . . . . .	25
fbGetPages . . . . .	31
fbGetSettings . . . . .	32
fbGetToken . . . . .	32
fbGetUserAdAccounts . . . . .	33
fbSetters . . . . .	34
fbUpdateAdAccountUsers . . . . .	34

<b>Index</b>	<b>36</b>
--------------	-----------

---

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/docs/marketing-api>>.

## Details

The DESCRIPTION file:

```
Package:      rfacebookstat
Type:        Package
Title:       Load Data from Facebook API Marketing
Version:     2.14.0
Authors@R:  c(person(given = "Alexey", family = "Seleznev", role = c("aut", "cre"), email = "selesnow@gmail.com", co
Maintainer:  Alexey Seleznev <selesnow@gmail.com>
Description: Load data by campaigns, ads, ad sets and insights, ad account and business manager from Facebook Marke
```

Depends: R ( $\geq$  4.1.0)  
 BugReports: <https://github.com/selesnow/rfacebookstat/issues>  
 Imports: httr, httr2, jsonlite, dplyr, purrr, tidyr ( $\geq$  1.0.0), stringr, tidyselct, pbapply  
 URL: <https://selesnow.github.io/rfacebookstat/>, <https://www.youtube.com/playlist?list=PLD2LDq8edf4pItOb-vZ>  
 License: MIT + file LICENSE  
 Suggests: knitr, rmarkdown  
 Encoding: UTF-8  
 RoxygenNote: 7.2.3  
 VignetteBuilder: knitr  
 Author: Alexey Seleznev [aut, cre] (<<https://orcid.org/0000-0003-0410-7385>>)

## Index of help topics:

fbAuth	Authorization in Facebook.
fbDeleteAdAccountUsers	Delete users from ad accounts.
fbGetAdAccountCustomAudiences	The Custom Audiences Associated With the Ad Account.
fbGetAdAccountUsers	Get User List From Account
fbGetAdAccountUsersPermissions	Get ad account user list with him permissions.
fbGetAdAccounts	Get available 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
fbGetBusinessManagers	Get available business Managers.
fbGetBusinessManagersUsers	Get a list of business manager users
fbGetBusinessUserAdAccounts	Get a list of accounts for a business manager user
fbGetCampaigns	Get campaign list from facebook marketing API
fbGetCatalogs	Get catalogs
fbGetCostData	Get facebook ads cost data.
fbGetLogins	Get all authorization logins
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.
fbSetters	Set rfacebookstat options

```
fbUpdateAdAccountUsers      Add users and update permission list.
rfacebookstat-package      Load Data from Facebook API Marketing
```

**Author(s)**

Alexey Seleznev [aut, cre] (<<https://orcid.org/0000-0003-0410-7385>>)  
 Maintainer: Alexey Seleznev <selesnow@gmail.com>

**See Also**

See oficial documantation: rfacebookstat package - <https://selesnow.github.io/rfacebookstat/> Facebook API Marketing - <https://developers.facebook.com/docs/marketing-api>

**Examples**

```
## Not run:
library(rfacebookstat)
# Auth
token <- fbGetToken(app_id = 00000000000000)

# 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 <a href="#">documentation</a>
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 0000000000000000 is id of your app in facebook
tk <- fbAuth(0000000000000000)
#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)
```

---

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_111111111111111111'.
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

*Get avable ad account list.*

---

**Description**

fbGetAdAccounts get data frame with avable in your bussines manager ad account list in 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"))
```

**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**

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

## 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_111111111111111111'.
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 = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

## 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

**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 = 0000000)
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"),
  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 < <a href="https://developers.facebook.com/docs/marketing-api/reference/adgroup/#">https://developers.facebook.com/docs/marketing-api/reference/adgroup/#</a> >.
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 = 00000000)
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_111111111111111111'.
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 = 0000000)
adsets <- fbGetAdSets(accounts_id = "act_111111111111111111",
                      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)
```

---

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 = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")  
  
## 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**

business_ids	IDs by your Business Manager
user_types	user types: business_users, system_users, pending_users.
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 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	<i>Get campaign list from facebook marketing API</i>
----------------	--

---

**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_111111111111111111'.
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 = 0000000)
camp <- fbGetCampaigns(accounts_id = "act_111111111111111111",
                       api_version = 'v3.0',
                       access_token = fb_token)

## End(Not run)
```

---

fbGetCatalogs

*Get catalogs*


---

**Description**

Load catalogs from business manager.

**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**

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**

```
## 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      *Get API facebook long time token.*

---

**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      *Get statistic by ad accounts.*

---

**Description**

fbGetMarketingStat is main function of rfacebookstat package, intended for load statistit 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 <a href="#">go</a> , 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':'/'} 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')`



---

fbGetPages	<i>Get pages list</i>
------------	-----------------------

---

### Description

Load data about all available 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

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 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 <a href="#">documentation</a>

**Value**

API token

**Examples**

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

## End(Not run)
```

---

fbGetUserAdAccounts    *User's ad account list.*

---

**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                      *Set rfacebookstat options*

---

### 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  
*Add users and update permission list.*

---

### 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 ( <a href="https://developers.facebook.com/docs/marketing-api/aduser/v2.12">https://developers.facebook.com/docs/marketing-api/aduser/v2.12</a> ).
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 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, 4

fbDeleteAdAccountUsers, 5

fbGetAdAccountCustomAudiences, 6

fbGetAdAccounts, 7

fbGetAdAccountsConversions, 8

fbGetAdAccountUsers, 9

fbGetAdAccountUsersPermissions, 10

fbGetAdCreative, 11

fbGetAds, 13

fbGetAdSets, 14

fbGetAdVideos, 16

fbGetApps, 17

fbGetBusinessManagers, 18

fbGetBusinessManagersUsers, 19

fbGetBusinessUserAdAccounts, 20

fbGetCampaigns, 21

fbGetCatalogs, 22

fbGetCostData, 23

fbGetLogins, 24

fbGetLongTimeToken, 25

fbGetMarketingStat, 25

fbGetPages, 31

fbGetSettings, 32

fbGetToken, 32

fbGetUserAdAccounts, 33

fbSetAccount (fbSetters), 34

fbSetApiVersion (fbSetters), 34

fbSetBusinessId (fbSetters), 34

fbSetters, 34

fbSetTokenPath (fbSetters), 34

fbSetUsername (fbSetters), 34

fbUpdateAdAccountUsers, 34

rfacebookstat (rfacebookstat-package), 2

rfacebookstat-package, 2