

# Package ‘rfacebookstat’

May 21, 2019

**Type** Package

**Title** Load Data from Facebook API Marketing

**Version** 1.9.0

**Date** 2019-05-20

**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** data.table, httr, RCurl, bitops, jsonlite, dplyr, purrr, tidy, stringr

**URL** <http://selesnow.github.io/rfacebookstat>

**License** GPL-2

**VignetteBuilder** knitr

**Suggests** knitr, rmarkdown

**Encoding** UTF-8

**Language** ru

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-05-21 19:40:03 UTC

## R topics documented:

rfacebookstat-package . . . . .	2
fbDeleteAdAccountUsers . . . . .	4
fbGetAdAccounts . . . . .	5
fbGetAdAccountUsers . . . . .	6

fbGetAdAccountUsersPermissions . . . . .	7
fbGetAdCreative . . . . .	8
fbGetAds . . . . .	9
fbGetAdSets . . . . .	10
fbGetApps . . . . .	12
fbGetBusinessManagers . . . . .	13
fbGetCampaigns . . . . .	13
fbGetCatalogs . . . . .	14
fbGetLongTimeToken . . . . .	15
fbGetMarketingStat . . . . .	16
fbGetPages . . . . .	20
fbGetProjects . . . . .	21
fbGetToken . . . . .	21
fbUpdateAdAccountUsers . . . . .	22

<b>Index</b>	<b>24</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-apis/>>.

## Details

The DESCRIPTION file:

```

Package:      rfacebookstat
Type:         Package
Title:        Load Data from Facebook API Marketing
Version:      1.9.0
Date:         2019-05-20
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
Depends:      R (>= 3.5.0)
BugReports:   https://github.com/selesnow/rfacebookstat/issues
Imports:      data.table, httr, RCurl, bitops, jsonlite, dplyr, purrr, tidyr, stringr
URL:          http://selesnow.github.io/rfacebookstat
License:      GPL-2
VignetteBuilder: knitr
Suggests:     knitr, rmarkdown
Encoding:     UTF-8
Language:     ru

```

## Index of help topics:

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.
fbGetAdCreative	Get creative list from facebook marketing API
fbGetAdSets	Get creative list from facebook marketing API
fbGetAds	Get ads list from facebook marketing API
fbGetApps	Get all the apps under a project
fbGetBusinessManagers	Get avable business Managers.
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
fbGetProjects	Get project list
fbGetToken	Get API facebook token.
fbUpdateAdAccountUsers	Add users and update permission list.
rfacebookstat-package	Load Data from Facebook API Marketing

Further information is available in the following vignettes:

rfacebookstat-get-statistics Vignette Title (source, pdf)

**Author(s)**

Alexey Seleznev <selesnow@gmail.com>

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-apis/>

**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)
```

---

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"),
  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.
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 <- fbGetToken(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)
```

---

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

---

**Description**

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

**Usage**

```
fbGetAdAccounts(source_id = getOption("rfacebookstat.business_id"),
                api_version = getOption("rfacebookstat.api_version"),
                access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

source_id	Your business manager or project ID.
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:
fbAccounts <- fbGetAdAccounts(source_id = "xxxxxxxxxxxxx",
                              api_version = "v2.10",
                              access_token = "xxxxxxx")

## 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",
  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"
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 = "xxxxxxxxxxxxxxxxxxxxxxxxxxxx")

## 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",  
  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>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"),
                api_version = getOption("rfacebookstat.api_version"),
                access_token = getOption("rfacebookstat.access_token"))
```

### Arguments

accounts_id	Your account ID, for example 'act_1111111111111111'.
api_version	Current Facebook API version.
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.  |
| 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"),
         api_version = getOption("rfacebookstat.api_version"),
         access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

accounts_id	Your account ID, for example 'act_1111111111111111'.
api_version	Current Facebook API version.
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**

```
## 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"),
            access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

- accounts\_id      Your account ID, for example 'act\_1111111111111111'.
- api\_version      Current Facebook API version.
- 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_1111111111111111",
                      api_version = 'v3.0',
                      access_token = fb_token)

## End(Not run)
```

---

fbGetApps

*Get all the apps under a project*


---

**Description**

Get data frame with list apps under a project.

**Usage**

```
fbGetApps(projects_id = NULL, api_version = getOption("rfacebookstat.api_version"),
          access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

projects_id	Your project ID.
api_version	Current Facebook API version.
access_token	Your facebook API token.

**Value**

Data frame with apps list.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
fbApps <- fbGetApps(projects_id = "xxxxxxxxxxxx",
                   api_version = "v2.8",
                   access_token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxx")

## 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"),
                      access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

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:
BM <- fbGetBusinessManagers(api_version = "v2.10", access_token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxx")

## 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"),
               access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

accounts\_id      Your account ID, for example 'act\_1111111111111111'.

api\_version      Current Facebook API version.

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_1111111111111111",
                        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"),
              access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

business\_id      ID by your Business Manager

api\_version      Current Facebook API version.

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)
```

---

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 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 = Sys.Date() - 30,
  date_stop = Sys.Date(),
  api_version = getOption("rfacebookstat.api_version"),
  action_report_time = NULL, interval = "day",
  console_type = "progressbar", request_speed = "normal",
  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 <a href="#">go</a> , 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 <a href="#">link</a> .
filtering	Vector of filtering or JSON string with array of filtering parameters, on example "ad.effective_status IN ARCHIVED" [{"field":"'ad.effective_status/'}, {"operator":"'and'"}] see filtering block for more examples
date_start	Start reporting date.
date_stop	End reporting day



request_speed	Speed beetwen 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
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 \*

### 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_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 = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX")

## End(Not run)
```

---

fbGetPages

*Get pages list*


---

**Description**

Load data about all available pages

**Usage**

```
fbGetPages(projects_id = NULL,
            api_version = getOption("rfacebookstat.api_version"),
            access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

projects\_id     Your project ID.  
 api\_version     Current Facebook API version.  
 access\_token    Your facebook API token.

**Value**

Data frame with pages list

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
fbPages <- fbGetPages(projects_id = "XXXXXXXXXXXXXXX",
                      api_version = "v2.8",
                      access_token = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX")

## End(Not run)
```

---

fbGetProjects	<i>Get project list</i>
---------------	-------------------------

---

**Description**

Load data about all available projects

**Usage**

```
fbGetProjects(bussiness_id = getOption("rfacebookstat.business_id"),
              api_version = getOption("rfacebookstat.api_version"),
              access_token = getOption("rfacebookstat.access_token"))
```

**Arguments**

bussiness_id	ID by your Business Manager
api_version	Current Facebook API version.
access_token	Your API facebook token

**Value**

Data frame with project list

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
fbProject <- fbGetProjects(bussiness_id = "xxxxxxxx",
                          access_token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxx")

## 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)
```

**Arguments**

app\_id            ID of your Facebook App

**Value**

API token

**Author(s)**

Alexey Seleznev

**Examples**

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

## End(Not run)
```

---

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"),
  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/docs/marketing-api/aduser/v2.12>).

access\_token     Your facebook API token.

user\_ids         vector with users ids who need remove from ad account.

api\_version      Current Facebook API version.



# Index

## \*Topic **package**

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

[fbDeleteAdAccountUsers](#), [4](#)

[fbGetAdAccounts](#), [5](#)

[fbGetAdAccountUsers](#), [6](#)

[fbGetAdAccountUsersPermissions](#), [7](#)

[fbGetAdCreative](#), [8](#)

[fbGetAds](#), [9](#)

[fbGetAdSets](#), [10](#)

[fbGetApps](#), [12](#)

[fbGetBusinessManagers](#), [13](#)

[fbGetCampaigns](#), [13](#)

[fbGetCatalogs](#), [14](#)

[fbGetLongTimeToken](#), [15](#)

[fbGetMarketingStat](#), [16](#)

[fbGetPages](#), [20](#)

[fbGetProjects](#), [21](#)

[fbGetToken](#), [21](#)

[fbUpdateAdAccountUsers](#), [22](#)

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

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