Package ‘fbRads’

August 21, 2023

Type Package
Title Analyzing and Managing Facebook Ads from R
Description Wrapper functions around the Facebook Marketing 'API' to create, read, update and delete custom audiences, images, campaigns, ad sets, ads and related content.
Version 17.0.0
Date 2023-07-25
URL https://github.com/daroczig/fbRads
License AGPL-3
Imports stats, RCurl, jsonlite, digest, logger, bit64, plyr, data.table
RoxygenNote 7.2.3
Encoding UTF-8
NeedsCompilation no
Author Gergely Daroczi [aut, cre], Ajaykumar Gopal [aut], Attila Nagy [aut], card.com [fnd], System1 [fnd]
Maintainer Gergely Daroczi <daroczig@rapporter.net>
Repository CRAN
Date/Publication 2023-08-20 23:12:37 UTC

R topics documented:

  fbad_add_audience ........................................ 3
  fbad_assign_users_to_account .......................... 3
  fbad_copy_ad ............................................. 4
  fbad_copy_adset ......................................... 5
  fbad_copy_campaign ................................. 6
  fbad_create_account ................................. 7
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>fbad_create_ad</td>
<td>8</td>
</tr>
<tr>
<td>fbad_create_adset</td>
<td>9</td>
</tr>
<tr>
<td>fbad_create_audience</td>
<td>10</td>
</tr>
<tr>
<td>fbad_create_campaign</td>
<td>11</td>
</tr>
<tr>
<td>fbad_create_creative</td>
<td>12</td>
</tr>
<tr>
<td>fbad_create_image</td>
<td>13</td>
</tr>
<tr>
<td>fbad_create_lookalike_audience</td>
<td>14</td>
</tr>
<tr>
<td>fbad_create_search</td>
<td>15</td>
</tr>
<tr>
<td>fbad_delete_audience</td>
<td>15</td>
</tr>
<tr>
<td>fbad_get_adaccounts</td>
<td>16</td>
</tr>
<tr>
<td>fbad_get_adaccount_details</td>
<td>16</td>
</tr>
<tr>
<td>fbad_get_client_ad_accounts</td>
<td>17</td>
</tr>
<tr>
<td>fbad_get_client_pages</td>
<td>17</td>
</tr>
<tr>
<td>fbad_get_my_ad_accounts</td>
<td>18</td>
</tr>
<tr>
<td>fbad_get_owned_ad_accounts</td>
<td>19</td>
</tr>
<tr>
<td>fbad_get_owned_pages</td>
<td>19</td>
</tr>
<tr>
<td>fbad_get_pixels</td>
<td>19</td>
</tr>
<tr>
<td>fbad_get_search</td>
<td>20</td>
</tr>
<tr>
<td>fbad_init</td>
<td>21</td>
</tr>
<tr>
<td>fbad_list_ad</td>
<td>22</td>
</tr>
<tr>
<td>fbad_list_adset</td>
<td>23</td>
</tr>
<tr>
<td>fbad_list_audience</td>
<td>23</td>
</tr>
<tr>
<td>fbad_list_campaign</td>
<td>24</td>
</tr>
<tr>
<td>fbad_preview_ad</td>
<td>25</td>
</tr>
<tr>
<td>fbad_reachestimate</td>
<td>25</td>
</tr>
<tr>
<td>fbad_read_ad</td>
<td>26</td>
</tr>
<tr>
<td>fbad_read_adset</td>
<td>27</td>
</tr>
<tr>
<td>fbad_read_audience</td>
<td>28</td>
</tr>
<tr>
<td>fbad_read_campaign</td>
<td>28</td>
</tr>
<tr>
<td>fbad_read_creative</td>
<td>29</td>
</tr>
<tr>
<td>fbad_remove_audience</td>
<td>30</td>
</tr>
<tr>
<td>fbad_share_audience</td>
<td>31</td>
</tr>
<tr>
<td>fbad_update_ad</td>
<td>31</td>
</tr>
<tr>
<td>fbad_update_adset</td>
<td>32</td>
</tr>
<tr>
<td>fbad_update_campaign</td>
<td>32</td>
</tr>
<tr>
<td>fbad_whoami</td>
<td>33</td>
</tr>
<tr>
<td>fbad_create_ad</td>
<td>33</td>
</tr>
<tr>
<td>fbad_create_adset</td>
<td>33</td>
</tr>
<tr>
<td>fbad_create_audience</td>
<td>33</td>
</tr>
<tr>
<td>fbad_create_campaign</td>
<td>33</td>
</tr>
<tr>
<td>fbad_create_creative</td>
<td>33</td>
</tr>
<tr>
<td>fbad_create_image</td>
<td>33</td>
</tr>
<tr>
<td>fbad_create_lookalike_audience</td>
<td>33</td>
</tr>
<tr>
<td>fbad_create_search</td>
<td>33</td>
</tr>
<tr>
<td>fbad_delete_audience</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_adaccounts</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_adaccount_details</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_client_ad_accounts</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_client_pages</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_my_ad_accounts</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_owned_ad_accounts</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_owned_pages</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_pixels</td>
<td>33</td>
</tr>
<tr>
<td>fbad_get_search</td>
<td>33</td>
</tr>
<tr>
<td>fbad_init</td>
<td>33</td>
</tr>
<tr>
<td>fbad_list_ad</td>
<td>33</td>
</tr>
<tr>
<td>fbad_list_adset</td>
<td>33</td>
</tr>
<tr>
<td>fbad_list_audience</td>
<td>33</td>
</tr>
<tr>
<td>fbad_list_campaign</td>
<td>33</td>
</tr>
<tr>
<td>fbad_preview_ad</td>
<td>33</td>
</tr>
<tr>
<td>fbad_reachestimate</td>
<td>33</td>
</tr>
<tr>
<td>fbad_read_ad</td>
<td>33</td>
</tr>
<tr>
<td>fbad_read_adset</td>
<td>33</td>
</tr>
<tr>
<td>fbad_read_audience</td>
<td>33</td>
</tr>
<tr>
<td>fbad_read_campaign</td>
<td>33</td>
</tr>
<tr>
<td>fbad_read_creative</td>
<td>33</td>
</tr>
<tr>
<td>fbad_remove_audience</td>
<td>33</td>
</tr>
<tr>
<td>fbad_share_audience</td>
<td>33</td>
</tr>
<tr>
<td>fbad_update_ad</td>
<td>33</td>
</tr>
<tr>
<td>fbad_update_adset</td>
<td>33</td>
</tr>
<tr>
<td>fbad_update_campaign</td>
<td>33</td>
</tr>
<tr>
<td>fbad_whoami</td>
<td>33</td>
</tr>
</tbody>
</table>

**Index** 37
fbad_add_audience

Add people to a custom FB audience

Description
Add people to a custom FB audience

Usage
fbad_add_audience(fbacc, audience_id, schema = c("EMAIL", "PHONE"), hashes)

Arguments
- fbacc: (optional) FB_Ad_account object, which defaults to the last returned object of fbad_init.
- audience_id: string
- schema: only two schema are supported out of the four: you can add/remove persons to/from a custom audience by e-mail addresses or phone numbers
- hashes: character vector of e-mail addresses or phone numbers to be transformed to hashes

References
https://developers.facebook.com/docs/marketing-api/reference/custom-audience/users/
#Creating

fbad_assign_users_to_account

Assign a user to an Ad Account

Description
Assign a user to an Ad Account

Usage
fbad_assign_users_to_account(
    account_id,
    access_token,
    tasks = c("MANAGE", "ADVERTISE", "ANALYZE"),
    user
)
Arguments

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>account_id</td>
<td>string</td>
</tr>
<tr>
<td>access_token</td>
<td>FB Ads API token</td>
</tr>
<tr>
<td>tasks</td>
<td>enum</td>
</tr>
<tr>
<td>user</td>
<td>id</td>
</tr>
</tbody>
</table>

**fbad_copy_ad**  
*Create a copy of an ad.*

Description

Create a copy of an ad.

Usage

```python
fbad_copy_ad(
    fbacc,  # (optional) FB_Ad_account object, which defaults to the last returned object of fbad_init.
    ad_id,  # (numeric string or integer) id of the ad that you want to copy
    adset_id = NULL,  # (numeric string or integer) id adset you want to create a copy of the ad into
    status_option = NULL,  # (string) enum ACTIVE, PAUSED, INHERITED_FROM_SOURCE
    rename_strategy = NULL,  # (string) enum DEEP_RENAME, ONLY_TOP_LEVEL_RENAME, NO_RENAME
    rename_prefix = NULL,  # (string) a prefix to copy names. Defaults to null if not provided
    rename_suffix = NULL,  # (string) a suffix to copy names. Defaults to null if not provided
    ...  # further arguments passed to the API endpoint
)
```

References

https://developers.facebook.com/docs/marketing-api/reference/adgroup/copies#Creating
Create a copy of an ad set. You can copy a maximum of 3 entities between ad set and ads.

**Usage**

```r
fbad_copy_adset(
    fbacc,
    adset_id,
    campaign_id = NULL,
    start_time = NULL,
    end_time = NULL,
    deep_copy = NULL,
    status_option = NULL,
    rename_strategy = NULL,
    rename_prefix = NULL,
    rename_suffix = NULL,
    ...
)
```

**Arguments**

- `fbacc` (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- `adset_id` (numeric string or integer) id adset you want to create a copy of
- `campaign_id` (numeric string or integer) Single ID of a campaign to make parent of the copy. Ignore if you want to keep the copy under the original campaign parent
- `start_time` (datetime) The start time of the ad set. If not set, the copied adset will inherit the start time from the original set
- `end_time` (datetime) The end time of the ad set
- `deep_copy` (boolean) Default value: false. Whether to copy all the child ads
- `status_option` (string) enum ACTIVE, PAUSED, INHERITED_FROM_SOURCE
- `rename_strategy` (string) enum DEEP_RENAME, ONLY_TOP_LEVEL_RENAME, NO_RENAME
- `rename_prefix` (string) a prefix to copy names. Defaults to null if not provided
- `rename_suffix` (string) a suffix to copy names. Defaults to null if not provided
- `...` further arguments passed to the API endpoint

**References**

https://developers.facebook.com/docs/marketing-api/reference/ad-campaign/copies/
**fbad_copy_campaign**

Create a copy of a campaign. You can copy a maximum of 3 entities between campaign, ad sets and ads.

**Description**

Create a copy of a campaign. You can copy a maximum of 3 entities between campaign, ad sets and ads.

**Usage**

```python
fbad_copy_campaign(
    fbacc,
    campaign_id,
    start_time = NULL,
    end_time = NULL,
    deep_copy = NULL,
    status_option = NULL,
    rename_strategy = NULL,
    rename_prefix = NULL,
    rename_suffix = NULL,
    ...
)
```

**Arguments**

- `fbacc`: (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`
- `campaign_id`: (numeric string or integer) Single ID of a campaign to make parent of the copy. Ignore if you want to keep the copy under the original campaign parent
- `start_time`: (datetime) The start time of the ad set. If not set, the copied adset will inherit the start time from the original set
- `end_time`: (datetime) The end time of the ad set
- `deep_copy`: (boolean) Default value: false. Whether to copy all the child ads
- `status_option`: (string) enum ACTIVE, PAUSED, INHERITED_FROM_SOURCE
- `rename_strategy`: (string) enum DEEP_RENAME, ONLY_TOP_LEVEL_RENAME, NO_RENAME
- `rename_prefix`: (string) a prefix to copy names. Defaults to null if not provided
- `rename_suffix`: (string) a suffix to copy names. Defaults to null if not provided
- `...`: further arguments passed to the API endpoint

**References**

[https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-group/copies/](https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-group/copies/)
fbad_create_account

Create Ad Account

Description
Create Ad Account

Usage

```python
cbad_create_account(
business_id,
access_token,
name,
currency,
timezone_id,
end_advertiser = "NONE",
media_agency = "NONE",
partner = "NONE",
...)
```

Arguments

- `business_id` : Business Manager id
- `access_token` : token
- `name` : string
- `currency` : ISO 4217 Currency Code
- `timezone_id` : https://developers.facebook.com/docs/marketing-api/reference/ad-account/timezone-ids
- `end_advertiser` : string
- `media_agency` : string
- `partner` : string
- ... further parameters passed to the Facebook API

Value

Ad Account id

References

https://developers.facebook.com/docs/marketing-api/reference/business/adaccount/#Creating
Create ad

Usage

fbad_create_ad(fbac, name, adset_id, creative_id, status = c("ACTIVE", "PAUSED"), ...)

Arguments

- fbacc: (optional) FB_Ad_account object, which defaults to the last returned object of fbad_init.
- name: Ad group name
- adset_id: Ad Set id
- creative_id: creative ID
- status: initial status of the Ad group
- ...: further parameters passed to the Facebook API

Value

ad id

References

https://developers.facebook.com/docs/marketing-api/reference/adgroup#Creating
fbad_create_adset  Create Ad Set

Description

Create Ad Set

Usage

fbad_create_adset(  
    fbacc,
    name,
    optimization_goal = c("NONE", "APP_INSTALLS", "CLICKS", "ENGAGED_USERS", "EXTERNAL",  
        "EVENT_RESPONSES", "IMPRESSIONS", "LINK_CLICKS", "OFFER_CLAIMS",  
        "OFFSITE_CONVERSIONS", "PAGE_ENGAGEMENT", "PAGELikes", "POST_ENGAGEMENT", "REACH",  
        "SOCIAL_IMPRESSIONS", "VIDEO_VIEWS"),
    billing_event = c("APP_INSTALLS", "CLICKS", "IMPRESSIONS", "LINK_CLICKS",  
        "OFFER_CLAIMS", "PAGE_LIKES", "POST_ENGAGEMENT", "VIDEO_VIEWS"),
    is_autobid = FALSE,
    bid_amount,
    promoted_object,
    campaign_id,
    status = c("ACTIVE", "PAUSED", "ARCHIVED", "DELETED"),
    daily_budget,
    lifetime_budget,
    end_time,
    start_time,
    targeting,
    ...
)

Arguments

fbacc  (optional) FB_Ad_account object, which defaults to the last returned object of  
    fbad_init.

name  name of the Ad Set

optimization_goal  optimization goal

billing_event  the billing event

is_autobid  logical. If TRUE, autobid is set and you do not need to specify bid_amount

bid_amount  integer

promoted_object  see at https://developers.facebook.com/docs/marketing-api/reference/  
    ad-promoted-object

campaign_id  parent Ad Campaign id
status          Ad Set status
daily_budget    using account currency
lifetime_budget using account currency
end_time        UTC UNIX timestamp
start_time      UTC UNIX timestamp
targeting       list
...             further arguments passed to the API endpoint

Value
Ad Set id

References
https://developers.facebook.com/docs/marketing-api/reference/ad-campaign#Creating

fbad_create_audience  Create a new FB custom audience

Description
Create a new FB custom audience

Usage
fbad_create_audience(
  fbacc,     
  name,      
  description, 
  ...
)

Arguments
fbacc   (optional) FB_Ad_account object, which defaults to the last returned object of fbad_init.
name    string
description optional string
subtype audience type
... any further parameters (fields) passed to the API
 Value

 custom audience ID

 References

 https://developers.facebook.com/docs/marketing-api/reference/custom-audience#Creating

\[
\text{fbad\_create\_campaign} \quad \text{Created Ad Campaign}
\]

 Description

 Created Ad Campaign

 Usage

 \[
 \text{fbad\_create\_campaign(}
 \begin{align*}
 \text{fbacc,} \\
 \text{buying\_type} &= \text{c("AUCTION", "FIXED\_CPM", "RESERVED")}, \\
 \text{campaign\_status} &= \text{c("ACTIVE", "PAUSED")}, \\
 \text{execution\_options} &= \text{NULL}, \\
 \text{name}, \\
 \text{objective} &= \text{c("BRAND\_AWARENESS", "CANVAS\_APP\_ENGAGEMENT", "CANVAS\_APP\_INSTALLS", \\
 \text{CONVERSIONS", "EVENT\_RESPONSES", "EXTERNAL", "LEAD\_GENERATION", "LINK\_CLICKS", \\
 "LOCAL\_AWARENESS", "MOBILE\_APP\_ENGAGEMENT", "MOBILE\_APP\_INSTALLS", "OFFER\_CLAIMS", \\
 "PAGE\_LIKES", "POST\_ENGAGEMENT", "PRODUCT\_CATALOG\_SALES", "VIDEO\_VIEWS"),} \\
 \text{spend\_cap} &= \text{NULL}
\end{align*}
\)
\]

 Arguments

 \[
 \text{fbacc} \quad \text{(optional) FB\_Ad\_account object, which defaults to the last returned object of fbad\_init.}
\]

 \[
 \text{buying\_type} \quad \text{Facebook optimization algorithm to delivery, pricing, and limits}
\]

 \[
 \text{campaign\_status} \quad \text{initial status of the Ad Campaign}
\]

 \[
 \text{execution\_options} \quad \text{special execution settings passed to the API}
\]

 \[
 \text{name} \quad \text{Ad Campaign name}
\]

 \[
 \text{objective} \quad \text{the campaign’s objective}
\]

 \[
 \text{spend\_cap} \quad \text{spend cap of the campaign}
\]

 Value

 Ad Campaign id
References

https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-group#

Creating

```r
fbad_create_creative

Create an ad creative
```

Description

Create an ad creative

Usage

```r
fbad_create_creative(
  fbacc,
  title,
  body,
  name,
  actor_id,
  follow_redirect,
  image_file,
  image_hash,
  image_crops,
  image_url,
  link_url,
  url_tags,
  object_id,
  object_story_id,
  object_story_spec,
  object_url,
  call_to_action_type = c("OPEN_LINK", "BOOK_TRAVEL", "SHOP_NOW", "PLAY_GAME",
                          "LISTEN_MUSIC", "WATCH_VIDEO", "USE_APP")
)
```

Arguments

- `fbacc` (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- `title` string
- `body` string
- `name` string
- `actor_id` Facebook object ID reference
- `follow_redirect` boolean
- `image_file` local image passed to Facebook. You might first upload the image via `fbad_create_image`. 
### fbad_create_image

**image_hash**  
string

**image_crops**  
list

**image_url**  
string

**link_url**  
string

**url_tags**  
list

**object_id**  
Facebook object ID reference

**object_story_id**  
post ID reference

**object_story_spec**  
list

**object_url**  
string list

**call_to_action_type**  
string

### Value

creative id

### Note

There are no checks done before passing provided arguments to Facebook. You have to know what you are up to. Read the Facebook docs.

### References

[https://developers.facebook.com/docs/marketing-api/reference/ad-creative#Creating](https://developers.facebook.com/docs/marketing-api/reference/ad-creative#Creating)

---

### fbad_create_image  Upload image

**Description**

Upload image

**Usage**

`fbad_create_image(fbac, img)`

**Arguments**

- **fbacc**  
  (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.

- **img**  
  file path
Value
list of file name, hash and URL

References
https://developers.facebook.com/docs/marketing-api/reference/ad-image#Creating

fbad_create_lookalike_audience

Create a new FB lookalike audience similar to an already existing custom audience

Description
Create a new FB lookalike audience similar to an already existing custom audience

Usage

```
fbad_create_lookalike_audience(
    fbacc,
    name,
    origin_audience_id,
    ratio = 0.01,
    country = "US"
)
```

Arguments

- **fbacc** *(optional)* FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- **name** string
- **origin_audience_id** numeric ID of origin custom audience
- **ratio** Between 0.01-0.20 and increments of 0.01. Indicates the top ratio percent of original audience in the selected country
- **country** Country name - the country to find the lookalike people.

Value
lookalike audience ID

References
https://developers.facebook.com/docs/marketing-api/audiences/guides/lookalike-audiences#create
Description
Delete a FB custom audience

Usage
fbad_delete_audience(fbacc, audience_id)

Arguments
fbacc  (optional) FB_Ad_account object, which defaults to the last returned object of fbad_init.
audience_id  numeric

Value
custom audience ID

References
https://developers.facebook.com/docs/marketing-api/reference/custom-audience#Deleting

Description
Deprecated in favor of fbad_get_owned_adaccounts

Usage
fbad_get_adaccounts(id, token, version, fields = c("name"), simplify = TRUE)

Arguments
id  Facebook Object, eg Ad Account (with act prefix) or a Business Manager Account ID
token  FB Ads API token (if running before fb_init)
version  Facebook Marketing API version (if running before fb_init)
fields  character vector
simplify  return data.frame or list
fbad_get_adaccount_details

*Get details for a Facebook Ads Account*

**Description**

Get details for a Facebook Ads Account

**Usage**

```r
fbad_get_adaccount_details(accountid, token, version)
```

**Arguments**

- `accountid`: Ads account graph object id
- `token`: FB Ads API token
- `version`: Facebook Marketing API version

**Value**

list(s) containing account details

**References**


---

fbad_get_client_ad_accounts

*Get account details of Ad Accounts belonging to the clients of a Business Manager Account*

**Description**

Get account details of Ad Accounts belonging to the clients of a Business Manager Account

**Usage**

```r
fbad_get_client_ad_accounts(
  id,
  token,
  version,
  fields = c("name"),
  simplify = TRUE
)
```
**fbad_get_client_pages**

*Get account details of Pages belonging to the clients of a Business Manager Account*

**Arguments**

- **id**: Facebook Object, eg Ad Account (with act prefix) or a Business Manager Account ID
- **token**: FB Ads API token (if running before fb_init)
- **version**: Facebook Marketing API version (if running before fb_init)
- **fields**: character vector
- **simplify**: return data.frame or list

**Usage**

`fbad_get_client_pages(id, token, version, fields = c("name"), simplify = TRUE)`

**fbad_get_my_ad_accounts**

*Get account details of Ad Accounts that are accessible by the given token*

**Description**

Get account details of Ad Accounts that are accessible by the given token

**Usage**

`fbad_get_my_ad_accounts(token, version)`
**Arguments**

- **token**  
  FB Ads API token (if running before `fb_init`)
- **version**  
  Facebook Marketing API version (if running before `fb_init`)

**Value**

character vector of Ad Account ids

---

**fbad_get_owned_ad_accounts**

*Get account details of Ad Accounts owned by a Business Manager Account*

---

**Description**

Get account details of Ad Accounts owned by a Business Manager Account

**Usage**

```r
fbad_get_owned_ad_accounts(
  id,
  token,
  version,
  fields = c("name"),
  simplify = TRUE
)
```

**Arguments**

- **id**  
  Facebook Object, eg Ad Account (with act prefix) or a Business Manager Account ID
- **token**  
  FB Ads API token (if running before `fb_init`)
- **version**  
  Facebook Marketing API version (if running before `fb_init`)
- **fields**  
  character vector
- **simplify**  
  return data.frame or list

**Value**

list(s) containing account details

**References**

[https://developers.facebook.com/docs/marketing-api/business-asset-management#adaccounts](https://developers.facebook.com/docs/marketing-api/business-asset-management#adaccounts)
**fbad_get_owned_pages**  Get account details of Pages owned by a Business Manager Account

**Description**

Get account details of Pages owned by a Business Manager Account

**Usage**

```
fbad_get_owned_pages(id, token, version, fields = c("name"), simplify = TRUE)
```

**Arguments**

- **id**
  - Facebook Object, eg Ad Account (with act prefix) or a Business Manager Account ID
- **token**
  - FB Ads API token (if running before fb_init)
- **version**
  - Facebook Marketing API version (if running before fb_init)
- **fields**
  - character vector
- **simplify**
  - return data.frame or list

**fbad_get_pixels**  Get tracking pixels of eg an Ad or Business Manager Account

**Description**

Get tracking pixels of eg an Ad or Business Manager Account

**Usage**

```
fbad_get_pixels(id, token, version, fields = c("name"), simplify = TRUE)
```

**Arguments**

- **id**
  - Facebook Object, eg Ad Account (with act prefix) or a Business Manager Account ID
- **token**
  - FB Ads API token
- **version**
  - Facebook Marketing API version
- **fields**
  - character vector
- **simplify**
  - return data.frame or list

**Value**

list(s) containing Ads Pixels
fbad_get_search

FB Search API Querying

Description

FB Search API Querying

Usage

```r
fbad_get_search(
  fbacc,
  q,
  type = c("adeducationschool", "adeducationmajor", "adgeolocation", "adcountry", "adzipcode", "adgeolocationmeta", "adradiussuggestion", "adinterest", "adinterestsuggestion", "adinterestvalid", "adlocale", "adTargetingCategory", "adworkemployer", "targetingsearch"),
  ...
)
```

Arguments

- `fbacc` (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- `q` string that is being searched for
- `type` describes the type of search eg: adinterest, adeducationmajor etc
- `...` other optional parameters accepted by the endpoint as key = value pairs eg: `limit = 5000`.

Value

data.frame containing results

References

https://developers.facebook.com/docs/marketing-api/audiences/reference/targeting-search

Examples

```r
## Not run:
fbad_get_search(q = "r programming language", type = "adinterest")
fbad_get_search(q = c("dog", "cat"), type = "adinterestvalid")
```

## End(Not run)
fbad_init

Initiate Facebook Account with OAuth token

Description

If you do not have a token, then register an (e.g. "Website") application at https://developers.facebook.com/apps and make a note of your "App ID" and "App Secret" at the "Dashboard" of your application. Then go to "Settings", click on "Add Platform", then "Website" and paste http://localhost:1410 as the "Site URL". Save, and then run the below example R commands to get your token. Please note that your app needs access to your ads as well, see https://developers.facebook.com/docs/marketing-api/overview/authorization for more details.

Usage

fbad_init(accountid, token, version = fb_api_most_recent_version())

Arguments

accountid  Facebook Ad account id without the act_prefix

token  Facebook OAuth token as a string

version  Facebook Marketing API version

Value

list returned invisibly containing versioned base URL and relevant API parameters

Examples

```r
## Not run:
## You can generate a token for future use with the help of `httr`, e.g.
library(httr)
app <- oauth_app("facebook", your_app_id, your_app_secret)
oauth2.0_token(oauth_endpoints("facebook"), app,
  scope = ",
  type = "application/x-www-form-urlencoded")$credentials$access_token

## Then pass this token with your account ID to fbad_init

## End(Not run)
```
fbad_list_ad  
*List all Ads for current account, list of Ad Sets or Campaigns*

**Description**

List all Ads for current account, list of Ad Sets or Campaigns

**Usage**

```
fbad_list_ad(fbacc, id, statuses, fields = "id", simplify = TRUE)
```

**Arguments**

- `fbacc` (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- `id` will do the look-up for all Ads based on this ID. Defaults to current FB account. Can be a (vector of) Ad Set or Campaign id(s).
- `statuses` character vector of statuses to use as a filter. Defaults to none. Please refer to the Facebook documentation for a list of possible values.
- `fields` character vector of fields to get from the API, defaults to `id`. Please refer to the Facebook documentation for a list of possible values.
- `simplify` boolean whether response is simplified to data.frame or else returned as raw list

**Value**

data.frame

**Note**

Will do a batched request to the Facebook API if multiple ids are provided.

**References**

[https://developers.facebook.com/docs/marketing-api/reference/adgroup#read-adaccount](https://developers.facebook.com/docs/marketing-api/reference/adgroup#read-adaccount)
**fbad_list_adset**

*List all Ad Sets for current account or Ad Campaign(s)*

**Description**

List all Ad Sets for current account or Ad Campaign(s)

**Usage**

```r
fbad_list_adset(fbacc, id, statuses, fields = "id", simplify = TRUE)
```

**Arguments**

- **fbacc** (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- **id** will do the look-up for all Ads based on this ID. Defaults to current FB account. Can be a (vector of) Ad Campaign id(s).
- **statuses** character vector of statuses to use as a filter. Defaults to none. Please refer to the Facebook documentation for a list of possible values.
- **fields** character vector of fields to get from the API, defaults to id. Please refer to the Facebook documentation for a list of possible values.
- **simplify** boolean whether response is simplified to data.frame or else returned as raw list

**References**

[https://developers.facebook.com/docs/marketing-api/reference/ad-campaign#Reading](https://developers.facebook.com/docs/marketing-api/reference/ad-campaign#Reading)

---

**fbad_list_audience**

*List all Custom Audiences for Ad account*

**Description**

List all Custom Audiences for Ad account

**Usage**

```r
fbad_list_audience(fbacc, id, statuses, fields = "id", simplify = TRUE)
```
### fbad_list_campaign

**Arguments**

- **fbacc** *(optional) FB_Ad_account object, which defaults to the last returned object of fbad_init.*
- **id** will do the look-up for all Ads based on this ID. Defaults to current FB account. Can be a (vector of) Ad Set or Campaign id(s).
- **statuses** character vector of statuses to use as a filter. Defaults to none. Please refer to the Facebook documentation for a list of possible values.
- **fields** character vector of fields to get from the API, defaults to id. Please refer to the Facebook documentation for a list of possible values.
- **simplify** boolean whether response is simplified to data.frame or else returned as raw list

**References**

https://developers.facebook.com/docs/marketing-api/reference/ad-account/customaudiences/

#Reading

---

### fbad_list_campaign  
*List all Ad Campaigns for current account*

**Description**

List all Ad Campaigns for current account

**Usage**

```r
fbad_list_campaign(fbacc, id, statuses, fields = "id", simplify = TRUE)
```

**Arguments**

- **fbacc** *(optional) FB_Ad_account object, which defaults to the last returned object of fbad_init.*
- **id** not supported argument
- **statuses** character vector of statuses to use as a filter. Defaults to none. Please refer to the Facebook documentation for a list of possible values.
- **fields** character vector of fields to get from the API, defaults to id. Please refer to the Facebook documentation for a list of possible values.
- **simplify** boolean whether response is simplified to data.frame or else returned as raw list

**References**

https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-group/

#Reading
### fbad_preview_ad

**Preview ad**

**Description**

Preview ad

**Usage**

```r
fbad_preview_ad(
  fbacc,
  id,
  ad_format = c("DESKTOP_FEED_STANDARD", "RIGHT_COLUMN_STANDARD", "MOBILE_FEED_STANDARD",
               "MOBILE_BANNER", "MOBILEInterstitial", "INSTAGRAM_STANDARD")
)
```

**Arguments**

- `fbacc` (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- `id` ad id(s)
- `ad_format` string

**References**

[https://developers.facebook.com/docs/marketing-api/generatepreview](https://developers.facebook.com/docs/marketing-api/generatepreview)

### fbad_reachestimate

**Query for reach estimate for given targeting spec**

**Description**

Query for reach estimate for given targeting spec

**Usage**

```r
fbad_reachestimate(fbacc, targeting_spec)
```

**Arguments**

- `fbacc` (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- `targeting_spec` lists of targeting spec characteristics as described at [https://developers.facebook.com/docs/marketing-api/audiences/reference/advanced-targeting](https://developers.facebook.com/docs/marketing-api/audiences/reference/advanced-targeting)
Value

list

References

https://developers.facebook.com/docs/marketing-api/reference/ad-account/reachestimate/

Examples

```r
## Not run:
library(jsonlite)
targetspec <- list(
  age_min = unbox(24),
  age_max = unbox(55),
  geo_locations = list(countries = 'US'))
fbad_reachestimate(targeting_spec = targetspec)
## End(Not run)
```

---

**fbad_read_ad**

**Read ad details**

Description

Read ad details

Usage

```r
fbad_read_ad(fbacc, id, fields = "id", simplify = TRUE)
```

Arguments

- `fbacc` (optional) FB_Ad_account object, which defaults to the last returned object of
  `fbad_init`.
- `id` ad id(s)
- `fields` character vector of fields to get from the API, defaults to id. Please refer to the Facebook documentation for a list of possible values.
- `simplify` return data.frame or list

Value

data.frame

Note

Will do a batched request to the Facebook API if multiple ids are provided.
fbad_read_adset

References

https://developers.facebook.com/docs/marketing-api/reference/adgroup#Reading

Examples

## Not run:
## get and Ad ID from yesterday
adid <- fb_insights(date_preset = 'yesterday', level = 'ad')[1]$ad_id[1]
## look for current status of the Ad
fbad_read_ad(id = adid, fields = c('effective_status'))

## End(Not run)

---

fbad_read_adset Read Ad Set details

Description

Read Ad Set details

Usage

fbad_read_adset(fbacc, id, fields = "id", simplify = TRUE)

Arguments

- **fbacc**: (optional) FB_Ad_account object, which defaults to the last returned object of fbad_init.
- **id**: ad set id(s)
- **fields**: character vector of fields to get from the API, defaults to id. Please refer to the Facebook documentation for a list of possible values.
- **simplify**: return data.frame or list

References

https://developers.facebook.com/docs/marketing-api/reference/ad-campaign#Reading
### fbad_read_audience

**Description**

Read metadata on a FB custom audience

**Usage**

```r
code

fbad_read_audience(fbac, audience_id, fields = c("id", "account_id", "approximate_count", "data_source", "delivery_status", "lookalike_audience_ids", "lookalike_spec", "name", "permission_for_actions", "operation_status", "subtype", "time_updated"))
```

**Arguments**

- `fbac` (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`
- `audience_id` numeric
- `fields` character vector of fields to be returned

**Value**

custom audience ID

**References**

[https://developers.facebook.com/docs/marketing-api/reference/custom-audience#Reading](https://developers.facebook.com/docs/marketing-api/reference/custom-audience#Reading)

---

### fbad_read_campaign

**Description**

Read Ad Campaign details

**Usage**

```r
fbad_read_campaign(fbac, id, fields = "id", simplify = TRUE)
```
Arguments

- **fbacc** (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- **id** Ad Campaign id(s)
- **fields** character vector of fields to get from the API, defaults to id. Please refer to the Facebook documentation for a list of possible values.
- **simplify** return data.frame or list

References

https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-group#

Reading

---

**fbad_read_creative**  
**Read ad creative**

Description

Read ad creative

Usage

```r
fbad_read_creative(
  fbacc,
  id,
  by = c("account", "creative", "ad"),
  fields = c("id", "name", "body", "title", "run_status", "actor_id",
             "call_to_action_type", "follow_redirect", "image_crops", "image_file", "image_hash",
             "image_url", "thumbnail_url", "link_url", "url_tags", "object_id", "object_story_id",
             "object_story_spec", "object_type", "object_url")
)
```

Arguments

- **fbacc** (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- **id** Ad Creative or Ad Set id
- **by** get all Ad Creatives for the account, for a given Ad or a single Creative
- **fields** character vector of fields to get from the API

Value

list to be post-processed
References

https://developers.facebook.com/docs/marketing-api/reference/ad-creative#Reading

Examples

```r
## Not run:
## get all creatives for the current account
fbad_read_creative(fbacc)

## End(Not run)
```

---

`fbad_remove_audience`  
*Add people from a custom FB audience*

Description

Add people from a custom FB audience

Usage

```r
fbad_remove_audience(fbacc, audience_id, schema = c("EMAIL", "PHONE"), hashes)
```

Arguments

- `fbacc` (optional) `FB_Ad_account` object, which defaults to the last returned object of `fbad_init`.
- `audience_id` string
- `schema` only two schema are supported out of the four: you can add/remove persons to/from a custom audience by e-mail addresses or phone numbers
- `hashes` character vector of e-mail addresses or phone numbers to be transformed to hashes

References

https://developers.facebook.com/docs/marketing-api/reference/custom-audience/users/#Deleting
`fbad_share_audience`  
*Share a FB custom audience with other accounts*

**Description**
Share a FB custom audience with other accounts

**Usage**

```r
fbad_share_audience(fbacc, audience_id, adaccounts)
```

**Arguments**

- `fbacc` (optional) `FB_Ad_account` object, which defaults to the last returned object of `fbad_init`.
- `audience_id` audience ID
- `adaccounts` numeric vector of FB account IDs

**Note**
This throws error if you provide wrong account ids OR even valid account ids that were previously granted access to the given custom audience.

---

`fbad_update_ad`  
*Update ad*

**Description**
Update ad

**Usage**

```r
fbad_update_ad(fbacc, id, ...)
```

**Arguments**

- `fbacc` (optional) `FB_Ad_account` object, which defaults to the last returned object of `fbad_init`.
- `id` ad id
- `...` parameters passed to Facebook API

**Value**
invisible TRUE
**fbad_update_campaign**

References

https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-group#Updating

---

**fbad_update_adset**  
*Update Ad Set*

Description

Update Ad Set

Usage

```python
fbad_update_adset(fbacc, id, ...)  
```

Arguments

- **fbacc**  
  (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- **id**  
  Ad Set id
- **...**  
  parameters passed to Facebook API

References

https://developers.facebook.com/docs/marketing-api/reference/adgroup#Updating

---

**fbad_update_campaign**  
*Update Ad Campaign*

Description

Update Ad Campaign

Usage

```python
fbad_update_campaign(fbacc, id, ...)  
```

Arguments

- **fbacc**  
  (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`.
- **id**  
  Ad Campaign id
- **...**  
  parameters passed to Facebook API

References

https://developers.facebook.com/docs/marketing-api/reference/ad-campaign#Updating
**fbad_whoami**  
*Prints user id and name*

**Description**
Prints user id and name

**Usage**

```
fbad_whoami(token, version)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>token</td>
<td>FB Ads API token (if running before <code>fb_init</code>)</td>
</tr>
<tr>
<td>version</td>
<td>Facebook Marketing API version (if running before <code>fb_init</code>)</td>
</tr>
</tbody>
</table>

**Value**

character vector of Ad Account ids

---

**fbRads**  
*fbRads package*

**Description**

This is a placeholder for storing import directives, please find more details in the README.md file of the package via `system.file` or on GitHub at [https://github.com/daroczig/fbRads](https://github.com/daroczig/fbRads).

---

**fb_api_most_recent_version**  
*Returns the most recent version of the supported Facebook Marketing API*

**Description**

Returns the most recent version of the supported Facebook Marketing API

**Usage**

```
fb_api_most_recent_version()
```

**Value**

string
**fb_api_version**

*Returns the currently used version of the Facebook Marketing API*

**Description**

Returns the currently used version of the Facebook Marketing API

**Usage**

```r
fb_api_version()
```

**Value**

string

---

**fb_insights**

*Insights*

**Description**

Insights

**Usage**

```r
fb_insights(
  fbacc,
  target = fbacc$acct_path,
  job_type = c("sync", "async"),
  retries = 0,
  simplify = TRUE,
  ...)
```

**Arguments**

- **fbacc** (optional) FB_Ad_account object, which defaults to the last returned object of `fbad_init`
- **target** ad account id(s) (default), campaign id(s), adset id(s) or ad id(s)
- **job_type** synchronous or asynchronous request. If the prior fails with "please reduce the amount of data", it will fall back to async request. Async query is possible with only one target.
- **retries** number of times this query has been sent to Facebook previously and failed – to be used internally for error handling
- **simplify** return a list of `data.frame` or `list` objects
- **...** named arguments passed to the API, like time range, fields, filtering etc.
fromJSONish

Value

list of data.frame or list objects depending on the simplify argument

References

https://developers.facebook.com/docs/marketing-api/insights

Examples

```r
## Not run:
fb_insights(fbacc)

## get default fields and filter for a few metrics
l <- fb_insights(date_preset = 'today', level = 'ad')
## merge pages into tabular data
data.table::rbindlist(l)

## sometimes more advanced list selections are needed
l <- fb_insights(date_preset = 'today', level = 'ad', fields = 'unique_actions', simplify = FALSE)

## filter for ACTIVE Ads
l <- fb_insights(date_preset = 'today', level = 'ad',
                  filtering = "[{'field': 'adset.effective_status', 'operator': 'IN', 'value': ['ACTIVE']}]")

## End(Not run)
```

fromJSONish

Validates and fixes some JSON issues, eg removing newlines etc

Description

Validates and fixes some JSON issues, eg removing newlines etc

Usage

fromJSONish(json, ...)

Arguments

- `json` string
- `...` passed to jsonlite

Value

parsed JSON as an R object
print.FB_Ad_Account  

Print method for custom fbRads class

Description

Print method for custom fbRads class

Usage

```r
## S3 method for class 'FB_Ad_Account'
print(x, ...)
```

Arguments

- `x`  
  R object with FB_Ad_Account class
- `...`  
  further arguments passed to print (ignored)
Index

fb_api_most_recent_version, 33
fb_api_version, 34
fb_insights, 34
fbad_add_audience, 3
fbad_assign_users_to_account, 3
fbad_copy_ad, 4
fbad_copy_adset, 5
fbad_copy_campaign, 6
fbad_create_account, 7
fbad_create_ad, 8
fbad_create_adset, 9
fbad_create_audience, 10
fbad_create_campaign, 11
fbad_create_creative, 12
fbad_create_image, 13
fbad_create_lookalike_audience, 14
fbad_delete_audience, 15
fbad_get_adaccount_details, 16
fbad_get_adaccounts, 15
fbad_get_client_ad_accounts, 16
fbad_get_client_pages, 17
fbad_get_my_ad_accounts, 17
fbad_get_owned_ad_accounts, 18
fbad_get_owned_pages, 19
fbad_get_pixels, 19
fbad_get_search, 20
fbad_init, 3–6, 8–15, 20, 21, 22–32, 34
fbad_list_ad, 22
fbad_list_adset, 23
fbad_list_audience, 23
fbad_list_campaign, 24
fbad_preview_ad, 25
fbad_reachestimate, 25
fbad_read_ad, 26
fbad_read_adset, 27
fbad_read_audience, 28
fbad_read_campaign, 28
fbad_read_creative, 29
fbad_remove_audience, 30
fbad_share_audience, 31
fbad_update_ad, 31
fbad_update_adset, 32
fbad_update_campaign, 32
fbad_whoami, 33
fbRads, 33
fromJSONish, 35
print.FB_Ad_Account, 36