

Package: cornucopia (via r-universe)

October 24, 2024

Title A cornucopia is like a funnel that keeps on giving

Version 0.0.0.9013

Description Facilitate reporting on sponsored and organic activities on Facebook, Instagram, and LinkedIn (currently), estimate and visualise the result of marketing funnels (long term)

License MIT + file LICENSE

Imports dplyr, purrr, fs, lubridate, httr2, cli, nomnomlgraph, shiny, golem, slider, RSQLite, googleAnalyticsR

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0), fbRads

Remotes giocomai/nomnomlgraph

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

VignetteBuilder knitr

Depends R (>= 2.10)

URL <https://giocomai.github.io/cornucopia/>

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

RemoteUrl <https://github.com/giocomai/cornucopia>

RemoteRef HEAD

RemoteSha d1028d448bebb8790f2aab932eca39bbcc2b0362

Contents

cc_api_get_fb_page_post_insights	3
cc_api_get_fb_video_insights	3
cc_api_get_instagram_media	4
cc_api_get_instagram_media_insights	5

cc_check_instagram_media_update	5
cc_date_to_json	6
cc_drive_upload_linkedin_stats_content	7
cc_drive_upload_linkedin_stats_followers	7
cc_drive_upload_linkedin_stats_visitors	8
cc_explorer	9
cc_get_fb_ads	10
cc_get_fb_ads_by_date	11
cc_get_fb_ads_daily	11
cc_get_fb_ads_totals_by_day	12
cc_get_fb_ad_actions_by_day	13
cc_get_fb_ad_campaigns	13
cc_get_fb_ad_creatives_id	14
cc_get_fb_ad_custom	15
cc_get_fb_ad_spend_totals_by_day_by_campaign	15
cc_get_fb_ad_spend_totals_by_day_by_objective	16
cc_get_fb_leads	17
cc_get_fb_long_page_token	18
cc_get_fb_long_user_token	19
cc_get_fb_managed_pages	20
cc_get_fb_owned_product_catalogs	20
cc_get_fb_page_insights	21
cc_get_fb_page_posts	22
cc_get_fb_page_post_insights	23
cc_get_fb_page_token	23
cc_get_fb_page_video	24
cc_get_fb_products	25
cc_get_fb_user	25
cc_get_fb_video_insights	26
cc_get_ga_event_ratio	27
cc_get_instagram_media	28
cc_get_instagram_media_id	29
cc_get_instagram_media_insights	29
cc_get_instagram_user	31
cc_get_instagram_user_id	32
cc_get_instagram_user_media	32
cc_get_linkedin_stats_content	33
cc_get_linkedin_stats_files	33
cc_get_linkedin_stats_followers	34
cc_get_linkedin_stats_visitors	34
cc_get_settings	35
cc_nomnoml	36
cc_read_fb_ads_by_date	36
cc_set	37
cc_ui_instagram_user_card	38
cc_valid_fields_ad_insights	39

`cc_api_get_fb_page_post_insights`

Get information about a single media directly from the API. Mostly used internally.

Description

See the official documentation for reference: <https://developers.facebook.com/docs/graph-api/reference/insights/#page-posts>

Usage

```
cc_api_get_fb_page_post_insights(  
    fb_post_id,  
    metrics = cc_valid_fields_fb_post_insights,  
    cache = TRUE,  
    update = TRUE,  
    api_version = "v19.0",  
    fb_page_id = NULL,  
    fb_page_token = NULL  
)
```

Arguments

<code>fb_post_id</code>	Instagram media identifier, must be a vector of length 1. A list of identifiers for your account can be retrieved with <code>'cc_get_fb_page_posts()'</code> .
<code>metrics</code>	Metrics to be retrieved. Consider that depending on the media type, different media types are effectively available. Requesting the wrong metrics will cause an error. Defaults to NULL. If left to NULL, metrics will be chosen based on the media type. See the official documentation for reference: https://developers.facebook.com/docs/graph-api/reference/insights/#page-posts
<code>cache</code>	Defaults to TRUE.

`cc_api_get_fb_video_insights`

Get information about a single Facebook video directly from the API. Mostly used internally.

Description

See the official documentation for reference: https://developers.facebook.com/docs/graph-api/reference/video/video_insights/

Usage

```
cc_api_get_fb_video_insights(
  fb_video_id,
  metrics = cc_valid_fields_fb_video_insights,
  cache = TRUE,
  update = TRUE,
  api_version = "v19.0",
  fb_page_id = NULL,
  fb_page_token = NULL
)
```

Arguments

fb_video_id	Instagram media identifier, must be a vector of length 1. A list of identifiers for your account can be retrieved with 'cc_get_fb_page_video()'.
metrics	Metrics to be retrieved. Consider that depending on the media type, different media types are effectively available. Requesting the wrong metrics will cause an error. Defaults to NULL. If left to NULL, metrics will be chosen based on the media type. See the official documentation for reference: https://developers.facebook.com/docs/graph-api/reference/insights/#page-posts
fb_page_token	

cc_api_get_instagram_media

Get information about a single media directly from the API. Mostly used internally.

Description

Get information about a single media directly from the API. Mostly used internally.

Usage

```
cc_api_get_instagram_media(
  ig_media_id,
  fields = cc_valid_fields_instagram_media_v,
  api_version = "v19.0",
  token = NULL
)
```

Arguments

ig_media_id	Instagram media identifier, must be a vector of length 1. A list of identifiers for your account can be retrieved with 'cc_get_instagram_media_id()'.
api_version	Defaults to "v18.0".

 cc_api_get_instagram_media_insights

Get information about a single media directly from the API. Mostly used internally.

Description

See the official documentation for reference: <https://developers.facebook.com/docs/instagram-api/reference/ig-media/insights>

Usage

```
cc_api_get_instagram_media_insights(
  ig_media_id,
  metrics = NULL,
  media_type = NULL,
  api_version = "v19.0",
  ig_user_id = NULL,
  token = NULL
)
```

Arguments

ig_media_id	Instagram media identifier, must be a vector of length 1. A list of identifiers for your account can be retrieved with 'cc_get_instagram_media_id()'
metrics	Metrics to be retrieved. Consider that depending on the media type, different media types are effectively available. Requesting the wrong metrics will cause an error. Defaults to NULL. If left to NULL, metrics will be chosen based on the media type. See the official documentation for reference: https://developers.facebook.com/docs/instagram-api/reference/ig-media/insights
media_type	Media type. Valid values include "IMAGE", "VIDEO", "REELS", and "CAROUSEL_ALBUM". Defaults to NULL. If not given, it will be retrieved with 'cc_get_instagram_media'. Ignored if 'metrics' explicitly given.
api_version	Defaults to "v18.0".
ig_user_id	Instagram user id, typically composed of 17 digits. Not to be confused with legacy Instagram account id.

 cc_check_instagram_media_update

Check when information about Instagram media should be updated based on a simple heuristic

Description

Update is set to TRUE if:

Usage

```
cc_check_instagram_media_update(
  ig_media_id = NULL,
  ig_user_id = NULL,
  insights = FALSE,
  token = NULL
)
```

Arguments

`ig_media_id` Instagram media identifier. A list of identifiers for your account can be retrieved with `'cc_get_instagram_media_id()'`. If left to `NULL`, a full list is automatically retrieved.

`ig_user_id` Instagram user id, typically composed of 17 digits. Not to be confused with legacy Instagram account id.

Details

- media has been published in the last week and not retrieved today - media has been published in the last month and not retrieved in the last week - media has been published in the last year and not retrieved in the last month - media has not been retrieved in the last year

Value

A tibble with four columns: `'ig_media_id'` (a character column), `'date_created'` and `'date_retrieved'` (Date columns), and an `'update'` column (logical)

<code>cc_date_to_json</code>	<i>Output date in a format that can be passed as custom date to the Facebook API.</i>
------------------------------	---

Description

Output date in a format that can be passed as custom date to the Facebook API.

Usage

```
cc_date_to_json(start_date, end_date = NULL)
```

Arguments

`start_date` A date.

`end_date` Defaults to `NULL`. If left to `NULL`, `'end_date'` is assumed to be the same as the start date

`cc_drive_upload_linkedin_stats_content`*Upload to Google Sheets Metrics statistics exported from LinkedIn*

Description

Upload to Google Sheets Metrics statistics exported from LinkedIn

Usage

```
cc_drive_upload_linkedin_stats_content(  
    path,  
    page_name,  
    sheet_dribble = NULL,  
    base_dribble = NULL,  
    export_csv = FALSE  
)
```

Arguments

<code>path</code>	Base path where all xls files exported from LinkedIn are stored.
<code>page_name</code>	Name of the page. If unsure about the exact form, see the relevant column of ‘ <code>cc_get_linkedin_stats_files()</code> ’
<code>sheet_dribble</code>	Dribble of the specific sheet where files should be uploaded.
<code>base_dribble</code>	Dribble of the folder where the relevant sheet is expected to be.
<code>export_csv</code>	Defaults to TRUE. If TRUE, exports content stats in a csv file in a folder with the same name as the base path, but with “_processed” appended.

`cc_drive_upload_linkedin_stats_followers`*Upload to Google Sheets New Followers statistics exported from LinkedIn*

Description

Upload to Google Sheets New Followers statistics exported from LinkedIn

Usage

```
cc_drive_upload_linkedin_stats_followers(  
    path,  
    page_name,  
    sheet_dribble = NULL,  
    base_dribble = NULL,  
    export_csv = FALSE  
)
```

Arguments

path	Base path where all xls files exported from LinkedIn are stored.
page_name	Name of the page. If unsure about the exact form, see the relevant column of 'cc_get_linkedin_stats_files()'
sheet_dribble	Dribble of the specific sheet where files should be uploaded.
base_dribble	Dribble of the folder where the relevant sheet is expected to be.
export_csv	Defaults to TRUE. If TRUE, exports follower stats in a csv file in a folder with the same name as the base path, but with "_processed" appended.

cc_drive_upload_linkedin_stats_visitors

Upload to Google Sheets Visitor metrics statistics exported from LinkedIn

Description

Upload to Google Sheets Visitor metrics statistics exported from LinkedIn

Usage

```
cc_drive_upload_linkedin_stats_visitors(
  path,
  page_name,
  sheet_dribble = NULL,
  base_dribble = NULL,
  export_csv = FALSE
)
```

Arguments

path	Base path where all xls files exported from LinkedIn are stored.
page_name	Name of the page. If unsure about the exact form, see the relevant column of 'cc_get_linkedin_stats_files()'
sheet_dribble	Dribble of the specific sheet where files should be uploaded.
base_dribble	Dribble of the folder where the relevant sheet is expected to be.
export_csv	Defaults to TRUE. If TRUE, exports visitors stats in a csv file in a folder with the same name as the base path, but with "_processed" appended.

Description

Run the Shiny Application

Usage

```
cc_explorer(
  start_date = cc_get_settings()[["start_date"]],
  end_date = cc_get_settings()[["end_date"]],
  fb_user_token = cc_get_settings()[["fb_user_token"]],
  fb_page_token = cc_get_settings()[["fb_page_token"]],
  fb_page_id = cc_get_settings()[["fb_page_id"]],
  fb_business_id = cc_get_settings()[["fb_business_id"]],
  fb_ad_account_id = cc_get_settings()[["fb_ad_account_id"]],
  fb_product_catalog_id = cc_get_settings()[["fb_product_catalog_id"]],
  fb_user_id = cc_get_settings()[["fb_user_id"]],
  ig_user_id = cc_get_settings()[["ig_user_id"]],
  ig_user_df = NULL,
  custom_head_html = "<meta name=\"referrer\" content=\"no-referrer\" />",
  onStart = NULL,
  options = list(),
  enableBookmarking = NULL,
  uiPattern = "/",
  ...
)
```

Arguments

start_date	Defaults to 91 days before today
end_date	Defaults to yesterday.
fb_user_token	Facebook user token (not a page token).
fb_business_id	Facebook business id. For details on how to find it, see https://www.facebook.com/business/help/1181250022022158
custom_head_html	Chunk of code to be included in the app's ' <code><head></code> '. This can be used, e.g., for custom analytics snippets. The default value, ' <code><meta name="referrer" content="no-referrer" /></code> ' asks the browser not to include the source website when following links to external websites.
onStart	A function that will be called before the app is actually run. This is only needed for <code>shinyAppObj</code> , since in the <code>shinyAppDir</code> case, a global <code>.R</code> file can be used for this purpose.

options	Named options that should be passed to the runApp call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify width and height parameters which provide a hint to the embedding environment about the ideal height/width for the app.
enableBookmarking	Can be one of "url", "server", or "disable". The default value, NULL, will respect the setting from any previous calls to <code>enableBookmarking()</code> . See <code>enableBookmarking()</code> for more information on bookmarking your app.
uiPattern	A regular expression that will be applied to each GET request to determine whether the ui should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.
...	arguments to pass to <code>golem_opts</code> . See <code>'?golem::get_golem_options'</code> for more details.

cc_get_fb_ads

Get details about Facebook ads

Description

Caches data in the folder `'fb_ads_by_date_rds'` in the current working directory.

Usage

```
cc_get_fb_ads(
  start_date = NULL,
  end_date = NULL,
  only_cached = FALSE,
  fields = c("campaign_name", "campaign_id", "adset_name", "adset_id", "ad_name",
            "ad_id", "objective", "account_currency", "spend", "actions", "action_values",
            "cost_per_action_type", "cost_per_unique_action_type", "conversions",
            "cost_per_conversion", "conversion_rate_ranking", "cpc", "cpm", "cpp", "ctr",
            "frequency", "reach")
)
```

Arguments

`only_cached` Defaults to FALSE. If TRUE, only pre-cached files within the given date range are loaded; no new calls to the API are made and reliably works offline.

Details

See also `'cc_get_fb_ads_by_date()'` for customisation of fields.

For valid fields, see: <https://developers.facebook.com/docs/marketing-api/reference/adgroup/insights/>

Value

A data frame with an extensive number of fields, some presented as nested data frames.

Examples

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

cc_get_fb_ads_by_date *Get Facebook ads by date and store them locally*

Description

Get Facebook ads by date and store them locally

Usage

```
cc_get_fb_ads_by_date(  
  date,  
  fields = c("campaign_name", "campaign_id", "adset_name", "adset_id", "ad_name",  
            "ad_id", "objective", "account_currency", "spend", "actions", "action_values",  
            "cost_per_action_type", "cost_per_unique_action_type", "conversions",  
            "cost_per_conversion", "conversion_rate_ranking", "cpc", "cpm", "cpp", "ctr",  
            "frequency", "reach")  
)
```

Arguments

date A vector of dates.

cc_get_fb_ads_daily *Get all campaigns for the current ad account / legacy*

Description

For details on fields, see <https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-group/>

Usage

```
cc_get_fb_ads_daily(
  fields = c("spend"),
  level = "campaign",
  start_date = NULL,
  end_date = NULL,
  api_version = "v19.0",
  fb_ad_account_id = NULL,
  max_pages = NULL,
  fb_user_token = NULL,
  cache = TRUE,
  update = TRUE
)
```

Arguments

`fields` A character vector of fields to retrieve. Defaults to "spend".

Details

It currently returns all fields that return a single value, 'cc_valid_fields_ad_campaign_group_v'

For reference, these are the fields that are always present for all campaigns: "id,name,created_time,updated_time,start_time,stop_time"

Cache updating currently suboptimal.

Examples

```
## Not run:
cc_get_fb_ads_daily()

## End(Not run)
```

```
cc_get_fb_ads_totals_by_day
```

Get total Facebook ad spending per day and as a rolling average

Description

Get total Facebook ad spending per day and as a rolling average

Usage

```
cc_get_fb_ads_totals_by_day(
  ads_df = NULL,
  start_date = NULL,
  end_date = NULL,
  before = 3,
  after = 3
)
```

Arguments

ads_df	A data frame, such as the one retrieved with 'cc_get_fb_ads()'
start_date	Defaults to 91 days before today
end_date	Defaults to yesterday.
before	Defaults to 3. Days to keep before the given day for calculating rolling averages.
after	Defaults to 3. Days to keep after the given day for calculating rolling averages.

 cc_get_fb_ad_actions_by_day

Takes all action results for all days when an ad is active

Description

For action breakdowns, see: <https://developers.facebook.com/docs/marketing-api/insights/breakdowns/>

Usage

```
cc_get_fb_ad_actions_by_day(
  ad_id = NULL,
  type = "actions",
  action_breakdowns = NULL,
  cache = FALSE,
  only_cached = FALSE,
  api_version = "v19.0",
  ad_account_id = NULL,
  token = NULL,
  max_pages = NULL
)
```

Details

Draft: caching disabled by default as only partly functional; Not yet fully tested with ads running longer than 25 days

 cc_get_fb_ad_campaigns

Get all campaigns for the current ad account / legacy

Description

For details on fields, see: <https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-group/>

Usage

```
cc_get_fb_ad_campaigns(
  fields = cc_valid_fields_ad_campaign_group_v,
  api_version = "v19.0",
  fb_ad_account_id = NULL,
  max_pages = NULL,
  token = NULL,
  cache = TRUE,
  update = TRUE
)
```

Arguments

fields A character vector of fields to retrieve. Defaults to all valid fields that return a single value, see: 'cc_valid_fields_ad_campaign_group_v'. Currently only default fields supported when caching.

Details

It currently returns all fields that return a single value, 'cc_valid_fields_ad_campaign_group_v'

For reference, these are the fields that are always present for all campaigns: "id,name,created_time,updated_time,start_time,stop_time"

Cache updating currently suboptimal.

Examples

```
## Not run:
cc_get_fb_ad_campaigns()

## End(Not run)
```

```
cc_get_fb_ad_creatives_id
```

Get Facebook ad creatives id for each ad id

Description

See: <https://developers.facebook.com/docs/marketing-api/reference/adgroup/adcreatives/>

Usage

```
cc_get_fb_ad_creatives_id(
  ad_id = NULL,
  ad_account_id = NULL,
  api_version = "v19.0",
  cache = TRUE,
  token = NULL
)
```

Value

A data frame with three character columns: 'ad_id', 'creative_id', and 'timestamp_retrieved'

cc_get_fb_ad_custom *Get custom stats about an ad (currently not functional)*

Description

<https://developers.facebook.com/docs/marketing-api/reference/ad-campaign-stats/>

Usage

```
cc_get_fb_ad_custom(  
  start_date = NULL,  
  end_date = NULL,  
  only_cached = FALSE,  
  api_version = "v19.0",  
  cache = TRUE,  
  ad_account_id = NULL,  
  token = NULL  
)
```

Arguments

token

Details

attribution window <https://developers.facebook.com/docs/marketing-api/reference/ads-action-stats/>

cc_get_fb_ad_spend_totals_by_day_by_campaign
Get total Facebook ad spending per day by campaign and as a rolling average

Description

Get total Facebook ad spending per day by campaign and as a rolling average

Usage

```
cc_get_fb_ad_spend_totals_by_day_by_campaign(  
  ads_df = NULL,  
  start_date = NULL,  
  end_date = NULL,  
  before = 3,  
  after = 3  
)
```

Arguments

ads_df	A data frame, such as the one retrieved with 'cc_get_fb_ads()'
start_date	Defaults to 91 days before today
end_date	Defaults to yesterday.
before	Defaults to 3. Days to keep before the given day for calculating rolling averages.
after	Defaults to 3. Days to keep after the given day for calculating rolling averages.

cc_get_fb_ad_spend_totals_by_day_by_objective

Get total Facebook ad spending per day by objective and as a rolling average

Description

Get total Facebook ad spending per day by objective and as a rolling average

Usage

```
cc_get_fb_ad_spend_totals_by_day_by_objective(  
  ads_df = NULL,  
  start_date = NULL,  
  end_date = NULL,  
  before = 3,  
  after = 3  
)
```

Arguments

ads_df	A data frame, such as the one retrieved with 'cc_get_fb_ads()'
start_date	Defaults to 91 days before today
end_date	Defaults to yesterday.
before	Defaults to 3. Days to keep before the given day for calculating rolling averages.
after	Defaults to 3. Days to keep after the given day for calculating rolling averages.

cc_get_fb_leads	<i>Retrieve leads from lead ads</i>
-----------------	-------------------------------------

Description

Currently does not cache results.

Usage

```
cc_get_fb_leads(
  form_id,
  fields = c("created_time", "id", "campaign_id", "campaign_name", "adset_id",
    "adset_name", "ad_id", "ad_name", "form_id", "is_organic", "platform", "field_data"),
  fb_page_token = NULL,
  fb_user_token = NULL,
  max_pages = NULL,
  api_version = "v19.0"
)
```

Arguments

form_id	Identifier of the lead gen form, presumably an id of about 17 digits.
fields	Defaults to 'c("created_time", "id", "ad_id", "form_id", "field_data")'-
fb_page_token	Takes precedence over 'fb_user_token'. See documentation for required permissions.
fb_user_token	Used as a fallback option if 'fb_page_token' not given. See documentation for required permissions.
max_pages	Maximum number of pages to retrieve (15 responses are included in each page). Defaults to 1000 pages internally.
api_version	Defaults to "v19.0".

Details

Details about the API: <https://developers.facebook.com/docs/marketing-api/guides/lead-ads/retrieving>

For the exact meaning of fields, see: <https://www.facebook.com/business/help/974763215942961?id=735435806665862>

Either a page or user token can be given. If both are given, page token is used.

In either case, be mindful of permission requirements:

To read ad specific fields, such as ad_id, campaign_id, you will need:

- A Page or User access token requested by a person who can advertise on the ad account and on the Page - The ads_management permission - The pages_read_engagement permission - The pages_show_list permission - The pages_manage_metadata permission - if using webhooks

To read all lead data and ad level data, you will need:

- A Page or User access token requested by a person who can advertise on the ad account and on the Page - The ads_management permission - The leads_retrieval permission - The pages_show_list permission - The pages_read_engagement permission - The pages_manage_ads permission

Value

A data frame

Examples

```
## Not run:  
cc_get_fb_leads(form_id = "12345678912345678")  
  
## End(Not run)
```

cc_get_fb_long_page_token

Get a long-lived page access token for Facebook

Description

For details, see <https://developers.facebook.com/docs/facebook-login/guides/access-tokens/get-long-lived/>

Usage

```
cc_get_fb_long_page_token(  
  fb_user_id = NULL,  
  fb_user_token = NULL,  
  api_version = "v19.0"  
)
```

Arguments

`fb_user_id` App-scoped user id. This can be retrieved with `'cc_get_fb_user()'`

`fb_user_token` Must be a long-lived user token. This can be retrieved with `'cc_get_fb_long_user_token()'`.

Details

You should use as input a long-lived user access token. Long-lived Page access token do not have an expiration date and only expire or are invalidated under certain conditions.

Value

A data frame with long-lived access tokens to all available pages.

Examples

```
## Not run:
if (interactive) {
  cc_get_fb_long_page_token(
    fb_user_id = "your_fb_user_id_here",
    fb_user_token = "your_long_term_token_here"
  )
}

## End(Not run)
```

cc_get_fb_long_user_token

Get a long-lived user access token for Facebook

Description

For details, see <https://developers.facebook.com/docs/facebook-login/guides/access-tokens/get-long-lived/>

Usage

```
cc_get_fb_long_user_token(
  fb_user_token = NULL,
  fb_app_id = NULL,
  fb_app_secret = NULL,
  api_version = "v19.0"
)
```

Details

You should use as input a short-lived user access token. In future calls, you can then routinely use your newly acquired long-lived user access token, which should generally last about 60 days.

Value

A character vector of length one, with the long-loved user access token.

Examples

```
## Not run:
if (interactive) {
  cc_get_fb_long_user_token(
    fb_user_token = "your_short_term_token_here",
    fb_app_id = "your_fb_app_id_here",
    fb_app_secret = "your_fb_app_secret_here"
  )
}
```

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

```
cc_get_fb_managed_pages
```

Get managed pages, including name, page token, and id

Description

Get managed pages, including name, page token, and id

Usage

```
cc_get_fb_managed_pages(  
  fields = c("id", "name"),  
  fb_user_id = NULL,  
  fb_user_token = NULL  
)
```

Arguments

fields	Defaults to 'c("id", "name")'. Include "access_token" in order to retrieve your Facebook page access token.
fb_user_id	Facebook user id. Defaults to NULL, can be set with 'cc_set()'. Can be retrieved with 'cc_get_fb_user()'.
fb_user_token	Facebook user token (not a page token).

Examples

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

```
cc_get_fb_owned_product_catalogs
```

Get information about your Facebook product catalogs

Description

See the relevant page in the documentation for available fields and more details <https://developers.facebook.com/docs/marketing-api/reference/product-catalog>

Usage

```
cc_get_fb_owned_product_catalogs(  
    fb_business_id = NULL,  
    api_version = "v19.0",  
    fields = c("id", "business", "commerce_merchant_settings", "da_display_settings",  
              "default_image_url", "fallback_image_url", "feed_count", "is_catalog_segment",  
              "name", "product_count", "vertical"),  
    token = NULL  
)
```

Arguments

`fb_business_id` Facebook business id. For details on how to find it, see <https://www.facebook.com/business/help/118125002202158>

`api_version` Defaults to "v18.0".

`fields` Defaults to all available.

`token` Facebook user token (not a page token).

Examples

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

```
cc_get_fb_page_insights  
Get Facebook page insights
```

Description

Official documentation: <https://developers.facebook.com/docs/graph-api/reference/v18.0/insights>

Usage

```
cc_get_fb_page_insights(  
    metric = c("page_impressions"),  
    start_date = NULL,  
    end_date = NULL,  
    api_version = "v19.0",  
    cache = TRUE,  
    fb_page_id = NULL,  
    fb_page_token = NULL  
)
```

Arguments

fb_page_token

cc_get_fb_page_posts *Get Facebook page post id and basic information that can be stored as strings*

Description

Find out what each of the fields effectively means in the official documentation: <https://developers.facebook.com/docs/graph-api/reference/v18.0/page/feed>

Usage

```
cc_get_fb_page_posts(
  api_version = "v19.0",
  max_pages = NULL,
  fields = names(cc_empty_fb_page_post_df),
  cache = TRUE,
  fb_page_id = NULL,
  fb_page_token = NULL
)
```

Arguments

max_pages	Posts are returned in pages of 25 posts each. How many pages should be retrieved? By default, this will try to retrieve all posts.
fields	Lists of fields which return data consistently, see ‘names(cc_empty_fb_page_post_df)’ for a full list and the official documentation for more details https://developers.facebook.com/docs/graph-api/reference/v18.0/page/feed . Expect caching to work consistently only if you leave this value unchanged. Consider disabling caching if you customise this parameter.
cache	Defaults to TRUE.

Value

A data frame, with the same columns as ‘cc_empty_fb_page_post_df’; each column in the returned data frame is of class character.

Examples

```
## Not run:
if (interactive) {
  fb_page_token <- cc_get_fb_page_token(
    fb_user_id = cc_get_fb_user(),
    page_name = "My example page"
  )
}
```

```
cc_set(fb_page_token = fb_page_token)
posts_df <- cc_get_fb_page_posts()
posts_df
}

## End(Not run)
```

cc_get_fb_page_post_insights *Get Facebook page posts insights*

Description

For reference, see: <https://developers.facebook.com/docs/graph-api/reference/insights/#page-posts>
defaults to Lifetime period for each post.

Usage

```
cc_get_fb_page_post_insights(
  fb_post_id = NULL,
  metrics = cc_valid_fields_fb_post_insights,
  cache = TRUE,
  update = TRUE,
  api_version = "v19.0",
  fb_page_id = NULL,
  fb_page_token = NULL
)
```

Arguments

fb_page_token

Details

Cache not yet working.

cc_get_fb_page_token *Get Facebook page token*

Description

Get Facebook page token

Usage

```
cc_get_fb_page_token(
  fb_user_id = NULL,
  page_id = NULL,
  page_name = NULL,
  fb_user_token = NULL
)
```

Arguments

fb_user_id	Facebook user id. If not known, can be retrieved with 'cc_get_fb_user()'.
page_id	Exact page id. See 'cc_get_fb_managed_pages()' for pages you have access to.
page_name	Exact page name. See 'cc_get_fb_managed_pages()' for pages you have access to.
token	User token.

Value

A character vector of length one with the Facebook token.

Examples

```
cc_get_fb_page_token(
  fb_user_id = cc_get_fb_user(),
  page_name = "My example page"
)
```

cc_get_fb_page_video *Get Facebook page video id and basic information that can be stored as strings*

Description

See: <https://developers.facebook.com/docs/video-api/guides/get-videos/>

Usage

```
cc_get_fb_page_video(
  api_version = "v19.0",
  max_pages = NULL,
  cache = TRUE,
  fb_page_id = NULL,
  fb_page_token = NULL
)
```

Arguments

fb_page_token

cc_get_fb_products *Get information about a Facebook catalog products*

Description

Data are not cached locally.

Usage

```
cc_get_fb_products(  
  fb_product_catalog_id = NULL,  
  fields = cc_valid_fields_fb_product,  
  api_version = "v19.0",  
  max_pages = NULL,  
  token = NULL  
)
```

Value

A data frame. Some columns include nested data.

Examples

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

cc_get_fb_user *Get Facebook user id and name, as well as other options fields*

Description

For details, see: <https://developers.facebook.com/docs/graph-api/reference/user/>

Usage

```
cc_get_fb_user(  
  fb_user_token = NULL,  
  fields = c("id", "name"),  
  format = "data.frame"  
)
```

Arguments

fb_user_token Facebook user token (not a page token).

Format

Defaults to "data.frame". If "list", a list is returned instead; useful e.g. when the "picture" field is requested.

Value

By default, a data frame with one row and two character columns, "name" and "id". Customisable with the 'format' argument.

Examples

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

```
cc_get_fb_video_insights  
      Get Facebook video insights
```

Description

For reference, see: https://developers.facebook.com/docs/graph-api/reference/video/video_insights/
defaults to Lifetime period for each video

Usage

```
cc_get_fb_video_insights(  
  fb_video_id,  
  metrics = cc_valid_fields_fb_video_insights,  
  cache = TRUE,  
  update = TRUE,  
  api_version = "v19.0",  
  fb_page_id = NULL,  
  fb_page_token = NULL  
)
```

Arguments

fb_page_token

cc_get_ga_event_ratio *Calculate the ratio of two Google Analytics events*

Description

Calculate the ratio of two Google Analytics events

Usage

```
cc_get_ga_event_ratio(  
  events,  
  rolling = FALSE,  
  rolling_before = 7,  
  rolling_after = 7,  
  rolling_period = "day",  
  rolling_complete = FALSE,  
  start_date = NULL,  
  end_date = NULL,  
  ga_email = NULL,  
  ga_property_id = NULL  
)
```

Arguments

events	A character vector of length 2, such as 'c("session_start", "purchase")' to calculate the ratio between these two events.
rolling	If TRUE, calculates a rolling mean over the number of periods (by default, days) set with 'rolling_before' and 'rolling_after'.
rolling_complete	Defaults to FALSE. If TRUE, rolling mean is calculated only for periods that are fully available.
start_date	Defaults to 91 days before today
end_date	Defaults to yesterday.
ga_email	E-mail address associated with relevant Google Analytics account, passed to 'googleAnalyticsR::ga_auth()'
ga_property_id	Google Analytics property identifier. Find it with 'googleAnalyticsR::ga_account_list("ga4")'.

Examples

```
## Not run:  
if (interactive) {  
  cc_get_ga_event_ratio(c("session_start", "purchase"))  
}  
  
## End(Not run)
```

 cc_get_instagram_media

Get details about a given Instagram post based on its ig_media_id

Description

It retrieves the requested fields from the APIs and introduces a few adjustments:

Usage

```
cc_get_instagram_media(
  ig_media_id = NULL,
  fields = cc_valid_fields_instagram_media_v,
  api_version = "v19.0",
  ig_user_id = NULL,
  update = TRUE,
  cache = TRUE,
  token = NULL
)
```

Arguments

ig_media_id	Instagram media identifier. A list of identifiers for your account can be retrieved with 'cc_get_instagram_media_id()'. If left to NULL, a full list is automatically retrieved.
api_version	Defaults to "v18.0".
ig_user_id	Instagram user id, typically composed of 17 digits. Not to be confused with legacy Instagram account id.

Details

- it always include the media id, in a column named 'ig_media_id' - if the field 'owner' is included, it outputs the result as a character vector (if more than one owner, separated by ;)
- it adds a 'timestamp_retrieved' column, with ISO 8601-formatted creation date in UTC
- it ensures that the output always included all requested fields, if they are valid; e.g. 'is_shared_to_feed' and 'media_url' may be omitted by the API (see documentation) but this function always includes the relevant column (and returns a NA value if no value is given)
- all valid fields for the given API endpoint are always requested and cached locally; only requested fields are effectively returned (but 'ig_media_id' and 'timestamp_retrieved' are always included as first and last column)

For details, see: <https://developers.facebook.com/docs/instagram-api/reference/ig-media>

Examples

```
## Not run:
cc_get_instagram_media()

## End(Not run)
```

`cc_get_instagram_media_id`*Get identifiers of all media published on a profile*

Description

Get identifiers of all media published on a profile

Usage

```
cc_get_instagram_media_id(  
  ig_user_id = NULL,  
  api_version = "v19.0",  
  token = NULL,  
  max_pages = NULL,  
  cache = TRUE  
)
```

Arguments

<code>ig_user_id</code>	Instagram user id, typically composed of 17 digits. Not to be confused with legacy Instagram account id.
<code>api_version</code>	Defaults to "v18.0".
<code>max_pages</code>	Defaults to NULL. If not given, it retrieves the number of media and calculates max number of pages needed. Hard-coded max number of pages at 1000.

Value

A tibble with one column named 'ig_media_id' with identifiers of Instagram media.

`cc_get_instagram_media_insights`*Get insights about a given Instagram post based on its ig_media_id*

Description

It retrieves the requested fields from the APIs and introduces a few adjustments:

Usage

```
cc_get_instagram_media_insights(  
    ig_media_id = NULL,  
    metrics = NULL,  
    api_version = "v19.0",  
    ig_user_id = NULL,  
    cache = TRUE,  
    update = TRUE,  
    token = NULL  
)
```

Arguments

<code>ig_media_id</code>	Instagram media identifier. A list of identifiers for your account can be retrieved with <code>'cc_get_instagram_media_id()'</code> . If left to <code>NULL</code> , a full list is automatically retrieved.
<code>api_version</code>	Defaults to "v18.0".
<code>ig_user_id</code>	Instagram user id, typically composed of 17 digits. Not to be confused with legacy Instagram account id.

Details

- it always includes the media id, in a column named `'ig_media_id'` - it always include the media type, in a column named `'ig_media_type'` - it adds a `'timestamp_retrieved'` column, with ISO 8601-formatted creation date in UTC - it ensures that the output always included all requested fields, if they are valid; e.g. `'is_shared_to_feed'` and `'media_url'` may be omitted by the API (see documentation) but this function always includes the relevant column (and returns a NA value if no value is given) - all valid fields for the given API endpoint are always requested and cached locally; only requested fields are effectively returned (but `'ig_media_id'` and `'timestamp_retrieved'` are always included as first and last column)

N.B.: different media types have different fields: hence the `'NA's` in columns for which data are unavailable for the given media type. N.B.: all media posted before 2017 are discarded by default, as Instagram API throw an error for earlier posts

For details, see: <https://developers.facebook.com/docs/instagram-api/reference/ig-media/insights>

Examples

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

cc_get_instagram_user *Get information about your Instagram user*

Description

If you need to retrieve your Instagram user id, consider 'cc_get_instagram_user_id()':

Usage

```
cc_get_instagram_user(  
  ig_user_id = NULL,  
  api_version = "v19.0",  
  fields = c("id", "ig_id", "username", "name", "biography", "website",  
            "followers_count", "follows_count", "media_count", "profile_picture_url",  
            "shopping_product_tag_eligibility"),  
  fb_user_token = NULL  
)
```

Arguments

ig_user_id	Instagram user id, typically composed of 17 digits. Not to be confused with legacy Instagram account id.
api_version	Defaults to "v18.0".
fields	Defaults to all available, consider reducing if you don't have all relevant permissions.
fb_user_token	Facebook user token (not a page token).

Details

See the relevant page in the documentation for available fields and more details <https://developers.facebook.com/docs/instagram-api/reference/ig-user>

Look in particular at the permissions requirements. If you have issues, consider dropping 'shopping_product_tag_eligibility' from the fields, as it requires additional permissions.

Examples

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

`cc_get_instagram_user_id`*Get the Instagram user id of a Page's Instagram Business Account*

Description

For details, see step 5 of this guide: <https://developers.facebook.com/docs/instagram-api/getting-started/>

Usage

```
cc_get_instagram_user_id(  
    fb_page_id = NULL,  
    fb_user_token = NULL,  
    api_version = "v19.0"  
)
```

Arguments

`fb_user_token` Facebook user token (not a page token).

`cc_get_instagram_user_media`*Retrieve information about other users through 'business_discovery'*

Description

Consider that only information about posts of creative or business users may be available. Given restrictions on the rate limit, you are likely to hit rate limits quite soon. Wait one hour and try again.

Usage

```
cc_get_instagram_user_media(  
    ig_username,  
    media_fields = c("like_count", "comments_count", "caption", "media_product_type",  
        "media_type", "media_url", "permalink", "thumbnail_url", "timestamp", "username"),  
    max_pages = NULL,  
    api_version = "v19.0",  
    ig_user_id = NULL,  
    fb_user_token = NULL  
)
```


Arguments

ig_username	A user name of an Instagram user.
media_fields	Defaults to all fields publicly available through 'business_discovery'.
max_pages	Posts are returned in pages of 25 posts each. How many pages should be retrieved? By default, this will try to retrieve all posts.

 cc_get_linkedin_stats_content

Get files with content stats exported from LinkedIn

Description

N.B. Only the "Metrics" sheet is processed, as other sheets give overall statistics, not limited to the relevant time period.

Usage

```
cc_get_linkedin_stats_content(path, page_name, export_csv = FALSE)
```

Arguments

path	Base path where all xls files exported from LinkedIn are stored.
page_name	Name of the page. If unsure about the exact form, see the relevant column of 'cc_get_linkedin_stats_files()'
export_csv	Defaults to TRUE. If TRUE, exports content stats in a csv file in a folder with the same name as the base path, but with "_processed" appended.

Value

A data frame.

 cc_get_linkedin_stats_files

Return a tibble with details about all the stats files exported from LinkedIn Pages

Description

Provide a path to a local folder where a bunch of files with names such as "pagename_followers_1684688073420.xls" are stored. The path will be scanned recursively and a tidy data frame returned, including columns with the name of the page, the type of stats included in each file, as well as the date when the data have been exported.

Usage

```
cc_get_linkedin_stats_files(path)
```

Arguments

path	A path to a folder to be scanned recursively
------	--

```
cc_get_linkedin_stats_followers
```

Get files with follower stats exported from LinkedIn

Description

N.B. Only the "New followers" sheet is processed, as other sheets give overall statistics, not limited to the relevant time period.

Usage

```
cc_get_linkedin_stats_followers(path, page_name, export_csv = FALSE)
```

Arguments

path	Base path where all xls files exported from LinkedIn are stored.
page_name	Name of the page. If unsure about the exact form, see the relevant column of 'cc_get_linkedin_stats_files()'
export_csv	Defaults to TRUE. If TRUE, exports follower stats in a csv file in a folder with the same name as the base path, but with "_processed" appended.

Value

A data frame.

```
cc_get_linkedin_stats_visitors
```

Get files with visitors stats exported from LinkedIn

Description

N.B. Only the "Metrics" sheet is processed, as other sheets give overall statistics, not limited to the relevant time period.

Usage

```
cc_get_linkedin_stats_visitors(path, page_name, export_csv = FALSE)
```

Arguments

path	Base path where all xls files exported from LinkedIn are stored.
page_name	Name of the page. If unsure about the exact form, see the relevant column of 'cc_get_linkedin_stats_files()'
export_csv	Defaults to TRUE. If TRUE, exports visitors stats in a csv file in a folder with the same name as the base path, but with "_processed" appended.

Value

A data frame.

cc_get_settings	<i>Retrieve settings for the session.</i>
-----------------	---

Description

Typically set with 'cc_set()'

Usage

```
cc_get_settings(
  start_date = NULL,
  end_date = NULL,
  fb_user_token = NULL,
  fb_page_token = NULL,
  fb_page_id = NULL,
  fb_business_id = NULL,
  fb_ad_account_id = NULL,
  fb_product_catalog_id = NULL,
  fb_user_id = NULL,
  ig_user_id = NULL,
  ga_email = NULL,
  ga_property_id = NULL
)
```

Arguments

start_date	Nominally defaults to NULL, but effectively to 91 days before today.
end_date	Nominally defaults to NULL, but effectively to yesterday.

Value

A list of named elements.

Examples

```
if (interactive()) {
  dates_1 <- cc_get_settings()
  dates_1
}
```

code_nomnoml

Show ad campaign structure in a nomnoml graph

Description

Show ad campaign structure in a nomnoml graph

Usage

```
cc_nomnoml(ads_df, direction = "right", drop_vertical_bar = FALSE, ...)
```

Arguments

`direction` Defaults to "right". Can be "down". Passed to 'nomnoml'.

`drop_vertical_bar` Defaults to FALSE. If TRUE, replaces vertical bar (which in 'nomnoml' can be used to separate a block in subsections) with "-".

...

code_read_fb_ads_by_date

Reads locally stored dated files, typically generated with 'cc_get_fb_ads_by_date'

Description

Reads locally stored dated files, typically generated with 'cc_get_fb_ads_by_date'

Usage

```
cc_read_fb_ads_by_date(
  path = "fb_ads_by_date_rds",
  start_date = NULL,
  end_date = NULL
)
```

Arguments

path	Path to dated files
start_date	Defaults to NULL. If given, only files retrieved on this date or later are included. Input should be of date class, or on the YYYY-MM-DD format.
end_date	Defaults to NULL. If given, only files retrieved on this date or sooner are included. Input should be of date class, or on the YYYY-MM-DD format.

Examples

```
## Not run:
if (interactive()) {
  cc_read_fb_ads_by_date()
}

## End(Not run)
```

cc_set	<i>Set settings and token for the session</i>
--------	---

Description

Set settings and token for the session

Usage

```
cc_set(
  start_date = NULL,
  end_date = NULL,
  fb_user_token = NULL,
  fb_page_token = NULL,
  fb_page_id = NULL,
  fb_business_id = NULL,
  fb_ad_account_id = NULL,
  fb_product_catalog_id = NULL,
  fb_user_id = NULL,
  ig_user_id = NULL,
  ga_email = NULL,
  ga_property_id = NULL
)
```

Arguments

start_date	Defaults to 91 days before today
end_date	Defaults to yesterday.
fb_user_token	Facebook user token (not a page token).

- fb_business_id Facebook business id. For details on how to find it, see <https://www.facebook.com/business/help/1181250022022158>
- ga_email E-mail address associated with relevant Google Analytics account, passed to 'googleAnalyticsR::ga_auth()'
- ga_property_id Google Analytics property identifier. Find it with 'googleAnalyticsR::ga_account_list("ga4")'.

Value

A list of named elements.

Examples

```
if (interactive()) {  
  dates_l <- cc_set()  
  dates_l  
}
```

cc_ui_instagram_user_card

Create a card to be used in dashboard or shiny app with basic information about the Instagram account

Description

Create a card to be used in dashboard or shiny app with basic information about the Instagram account

Usage

```
cc_ui_instagram_user_card(  
  ig_user_df = NULL,  
  ig_user_id = NULL,  
  fb_user_token = NULL  
)
```

Arguments

- ig_user_id Instagram user id, typically composed of 17 digits. Not to be confused with legacy Instagram account id.
- fb_user_token Facebook user token (not a page token).

Examples

```
## Not run:
if (interactive) {
  cc_ui_instagram_user_card()
}

## End(Not run)
```

cc_valid_fields_ad_insights

A list with all valid fields for the Ad Insights Marketing API

Description

As all fields cannot be requested at the same time, they are grouped here according to their thematic scope in a named list. As this division is not formalised in the official API, the named list has been liberally built by the package author. As the actual contents of some fields is not immediately obvious nor clearly described in the official documentation they may be misplaced until they are fully tested. Fields may be repeated when relevant. 'ad_id' is included in all lists as it can be used for matching.

Usage

```
cc_valid_fields_ad_insights
```

Format

```
## 'cc_valid_fields_ad_insights' A named list with all valid fields for Ad Insights.
```

about_the_account Retrieves information about the ad account. If you separate clearly your ad account, you probably don't need to request this information with every call.

about_the_ad Retrieves information about a given ad. Logically, these should need to be requested only once per ad and, e.g. not on a daily basis as they should remain unchanged for each ad.

about_the_ad_timing Retrieves information about the timing of a given ad, when it started, ended, etc. These also need to be requested only once per ad, but may need to be updated while the ad is still running.

Source

```
<https://developers.facebook.com/docs/marketing-api/reference/adgroup/insights/>
```

Index

* datasets

- [cc_valid_fields_ad_insights, 39](#)
- [cc_api_get_fb_page_post_insights, 3](#)
- [cc_api_get_fb_video_insights, 3](#)
- [cc_api_get_instagram_media, 4](#)
- [cc_api_get_instagram_media_insights, 5](#)
- [cc_check_instagram_media_update, 5](#)
- [cc_date_to_json, 6](#)
- [cc_drive_upload_linkedin_stats_content, 7](#)
- [cc_drive_upload_linkedin_stats_followers, 7](#)
- [cc_drive_upload_linkedin_stats_visitors, 8](#)
- [cc_explorer, 9](#)
- [cc_get_fb_ad_actions_by_day, 13](#)
- [cc_get_fb_ad_campaigns, 13](#)
- [cc_get_fb_ad_creatives_id, 14](#)
- [cc_get_fb_ad_custom, 15](#)
- [cc_get_fb_ad_spend_totals_by_day_by_campaign, 15](#)
- [cc_get_fb_ad_spend_totals_by_day_by_objective, 16](#)
- [cc_get_fb_ads, 10](#)
- [cc_get_fb_ads_by_date, 11](#)
- [cc_get_fb_ads_daily, 11](#)
- [cc_get_fb_ads_totals_by_day, 12](#)
- [cc_get_fb_leads, 17](#)
- [cc_get_fb_long_page_token, 18](#)
- [cc_get_fb_long_user_token, 19](#)
- [cc_get_fb_managed_pages, 20](#)
- [cc_get_fb_owned_product_catalogs, 20](#)
- [cc_get_fb_page_insights, 21](#)
- [cc_get_fb_page_post_insights, 23](#)
- [cc_get_fb_page_posts, 22](#)
- [cc_get_fb_page_token, 23](#)
- [cc_get_fb_page_video, 24](#)
- [cc_get_fb_products, 25](#)
- [cc_get_fb_user, 25](#)
- [cc_get_fb_video_insights, 26](#)
- [cc_get_ga_event_ratio, 27](#)
- [cc_get_instagram_media, 28](#)
- [cc_get_instagram_media_id, 29](#)
- [cc_get_instagram_media_insights, 29](#)
- [cc_get_instagram_user, 31](#)
- [cc_get_instagram_user_id, 32](#)
- [cc_get_instagram_user_media, 32](#)
- [cc_get_linkedin_stats_content, 33](#)
- [cc_get_linkedin_stats_files, 33](#)
- [cc_get_linkedin_stats_followers, 34](#)
- [cc_get_linkedin_stats_visitors, 34](#)
- [cc_get_settings, 35](#)
- [cc_nomnoml, 36](#)
- [cc_read_fb_ads_by_date, 36](#)
- [cc_set, 37](#)
- [cc_ui_instagram_user_card, 38](#)
- [cc_valid_fields_ad_insights, 39](#)

[enableBookmarking\(\), 10](#)