Package ‘rsocialwatcher’

May 6, 2024

Title  ‘Facebook Marketing API’ Social Watcher
Version  0.1.1
Description
Facilitates querying data from the ‘Facebook Marketing API’, particularly for social science research <https://developers.facebook.com/docs/marketing-apis/> . Data from the ‘Facebook Marketing API’ has been used for a variety of social science applications, such as for poverty estimation (Marty and Duhaut (2024) <doi:10.1038/s41598-023-49564-6> ), disease surveillance (Araujo et al. (2017) <doi:10.48550/arXiv.1705.04045> ), and measuring migration (Alexander, Polimis, and Zagheni (2020) <doi:10.1007/s11113-020-09599-3> ). The package facilitates querying the number of Facebook daily/monthly active users for multiple location types (e.g., from around a specific coordinate to an administrative region) and for a number of attribute types (e.g., interests, behaviors, education level, etc). The package supports making complex queries within one API call and making multiple API calls across different locations and/or parameters.

License  MIT + file LICENSE
Encoding  UTF-8
RoxygenNote  7.3.1
URL  https://worldbank.github.io/rsocialwatcher/
BugReports  https://github.com/worldbank/rsocialwatcher/issues
Suggests  knitr, rmarkdown
Imports  dplyr, lubridate, jsonlite, httr, curl, stringr, splitstackshape, sf, purrr
NeedsCompilation  no
Author  Robert Marty [aut, cre] (<https://orcid.org/0000-0002-3164-3813>)
Maintainer  Robert Marty <rmarty@worldbank.org>
Repository  CRAN
Date/Publication  2024-05-06 18:30:02 UTC
**R topics documented:**

- `get_fb_parameter_ids` ........................................... 2
- `get_fb_suggested_radius` .................................... 4
- `get_location_coords` .......................................... 5
- `map_param` ......................................................... 6
- `map_param_vec` ................................................... 7
- `query_fb_marketing_api` ....................................... 8

**Index**

---

### `get_fb_parameter_ids`  
*Get Facebook Parameter IDs*

**Description**

This function returns dataframes of Facebook parameters and their associated IDs for different categories of information. Categories include behaviors, interests, locales, job titles, education major, and location (e.g., country, city, zip code, etc). The returned dataframe contains ids that can be used in the `query_fb_marketing_api` function.

**Usage**

```r
get_fb_parameter_ids(
  type,
  version,
  token,
  q = NULL,
  country_code = NULL,
  region_id = NULL,
  key = NULL,
  limit = NULL,
  add_location_coords = FALSE
)
```

**Arguments**

- **type**  

- **version**  
  'Facebook Marketing' API version; for example, "v19.0"

- **token**  
  'Facebook Marketing' API token

- **q**  
  Query string to limit search. For example, when searching job titles, setting `q="data"` will return jobs with "data" in the name, such as "data science."
get_fb_parameter_ids

- `country_code`: When searching locations, limit the search to a specific country; for example, only search for cities within a specific country.
- `region_id`: When searching locations, limit the search to a specific region; for example, only search for cities within a specific region.
- `key`: When searching locations, limit the search to a specific location key; for example, only search for neighborhood within a specific city.
- `limit`: Number of parameter IDs to search for.
- `add_location_coords`: When querying location IDs (e.g., when `type = "city"`), add location coordinates—which will add the latitude and longitude, as well as the geometry when available. (Default: FALSE)

**Details**

For additional information, see: https://developers.facebook.com/docs/marketing-api/audiences/reference/targeting-search/

**Value**

Dataframe with parameter IDs and descriptions.

**Examples**

```r
## Not run:

### Define version and token
VERSION = "enter-version"
TOKEN = "enter-token"

### Query parameter IDs
get_fb_parameter_ids(type = "interests", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "behaviors", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "education_majors", q = "Computer", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "education_schools", q = "Washington", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "education_statuses", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "family_statuses", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "income", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "industries", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "work_positions", q = "Data", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "work_employers", q = "World Bank", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "relationship_statuses", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "life_events", version = VERSION, token = TOKEN)

### Location IDs
get_fb_parameter_ids(type = "country", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "region", country_code = "US", version = VERSION, token = TOKEN)
get_fb_parameter_ids(type = "medium_geo_area",
  country_code = "US",
  q = "Henrico",
  version = VERSION,
  ...)
```
### get_fb_suggested_radius

#### Get Suggested Radius

**Description**

When targeting around a specific location, returns a suggested radius to reach enough people.

**Usage**

```r
get_fb_suggested_radius(location, distance_unit = "kilometer", version, token)
```

**Arguments**

- `location` Vector of latitude and longitude (`c(lat, lon)`).
- `distance_unit` Either "kilometer" or "mile"; defaults to "kilometer".
- `version` 'Facebook Marketing' API version; for example, "v19.0".
- `token` 'Facebook Marketing' API token.

**Details**

For more information, see the Facebook documentation [here](#).

**Value**

Dataframe with suggested radius and distance unit.

**Examples**

```r
## Not run:
get_fb_suggested_radius(location = c(38.89831, -77.03658),
version = "v14.0",
token = "TOKEN-HERE")

## End(Not run)
```
**get_location_coords**  
*Get Coordinates/Geometries for Valid Location Keys*

**Description**

Get Coordinates/Geometries for Valid Location Keys

**Usage**

```r
going {  
  location_unit_type,  
  location_keys,  
  version,  
  token,  
  large_query_chunk_size = 10,  
  large_query_pause = 0,  
  limit = NULL,  
  verbose = TRUE  
}

get_location_coords(  
  location_unit_type,  
  location_keys,  
  version,  
  token,  
  large_query_chunk_size = 10,  
  large_query_pause = 0,  
  limit = NULL,  
  verbose = TRUE  
)
```

**Arguments**

- **location_unit_type**
  Either "coordinates" (for buffer around single point) or type of geographic location, including: "countries", "regions", "cities", "zips", "places", "geo_markets", "electoral_district", or "country_groups". See the Basic Targetting documentation for more information.

- **location_keys**
  Key associated with location. Use the `get_fb_parameter_ids` function to get location keys; see here for examples.

- **version**
  API version. e.g., "v19.0"

- **token**
  Facebook API token

- **large_query_chunk_size**
  The function will first try to query all locations using one API call. If too many locations are requested, the function will query in chunks. By default, the function will query 10 locations at a time. (Default: 10).

- **large_query_pause**
  The function will first try to query all locations using one API call. If too many locations are requested, the function will query in chunks. After each query, the `large_query_pause` can be set to > 0 to sleep for `large_query_pause` seconds in order to not make too many API calls too quickly. (Default: 0).

- **limit**
  Number of parameter IDs to search for.

- **verbose**
  If the function needs to make multiple queries to obtain location information for all location keys, print progress. (Default: TRUE).
Value

Spatial features dataframe

Examples

```r
## Not run:
#### Define version, creation act, and token
VERSION = "enter-version"
TOKEN = "enter-token"

#### Grab locations
loc_sf <- get_location_coords(location_type = "countries",
    location_keys = c("US", "MX", "CA"),
    version = VERSION,
    token = TOKEN)

## End(Not run)
```

Description

Map Parameters within `query_fb_marketing_api()` Groups input into a map_param object. When a map_param object is entered as a parameter in `query_fb_marketing_api()`, `query_fb_marketing_api()` makes a separate API query for each item within map_param. A map_param object is structured as a list, where the map_param class triggers the `query_fb_marketing_api` function to make a separate API query for each item in the list.

Usage

```r
map_param(...)
```

Arguments

```r
... Vector or list
```

Value

Object of class `map_param` to be used as input to the `query_fb_marketing_api()` function to make multiple API queries.
### Examples

```r
## Not run:
# Make 3 queries:
# 1. Number of males and females MAU/DAU
# 2. Number of male MAU/DAU
# 3. Number of female MAU/DAU
query_fb_marketing_api(
  location_unit_type = "countries",
  location_keys = "US",
  gender = map_param(c(1,2), 1, 2),
  version = VERSION,
  creation_act = CREATION_ACT,
  token = TOKEN)
## End(Not run)
```

---

**map_param_vec**  
*Map Parameters over Vector* Converts a vector into a `map_param` object. When a `map_param` object is entered as a parameter in `query_fb_marketing_api()`, `query_fb_marketing_api()` makes a separate API query for each item within `map_param` (i.e., for each item in the original vector entered into `map_param`). A `map_param` object is structured as a list, where the `map_param` class triggers the `query_fb_marketing_api` function to make a separate API query for each item in the list.

---

**Description**

Map Parameters over Vector Converts a vector into a `map_param` object. When a `map_param` object is entered as a parameter in `query_fb_marketing_api()`, `query_fb_marketing_api()` makes a separate API query for each item within `map_param` (i.e., for each item in the original vector entered into `map_param`). A `map_param` object is structured as a list, where the `map_param` class triggers the `query_fb_marketing_api` function to make a separate API query for each item in the list.

**Usage**

`map_param_vec(...)`

**Arguments**

`...`  
*Vector*

**Value**

Object of class `map_param` to be used as input to the `query_fb_marketing_api()` function to make multiple API queries.
Examples

## Not run:
# Make 2 queries:
# 1. Number of male MAU/DAU
# 2. Number of female MAU/DAU
query_fb_marketing_api(
  location_unit_type = "countries",
  location_keys = "US",
  gender = map_param_vec(1:2),
  version = VERSION,
  creation_act = CREATION_ACT,
  token = TOKEN)
## End(Not run)

### Description

Query 'Facebook Marketing' API

### Usage

query_fb_marketing_api(
  location_unit_type,
  lat_lon = NULL,
  radius = NULL,
  radius_unit = NULL,
  location_keys = NULL,
  location_types = c("home", "recent"),
  locales = NULL,
  interests = NULL,
  behaviors = NULL,
  college_years = NULL,
  education_majors = NULL,
  education_schools = NULL,
  education_statuses = NULL,
  family_statuses = NULL,
  income = NULL,
  industries = NULL,
  life_events = NULL,
  relationship_statuses = NULL,
  work_positions = NULL,
  work_employers = NULL,
  excl_interests = NULL,
excl_behaviors = NULL,
excl_college_years = NULL,
excl_education_majors = NULL,
excl_education_schools = NULL,
excl_education_statuses = NULL,
excl_family_statuses = NULL,
excl_income = NULL,
excl_industries = NULL,
excl_life_events = NULL,
excl_relationship_statuses = NULL,
excl_work_positions = NULL,
excl_work_employers = NULL,
user_os = NULL,
wireless_carrier = NULL,
gender = c(1, 2),
age_min = 18,
age_max = 65,
flex_target = NULL,
version,
token,
sleep_time = 0.1,
show_result = FALSE,
verbose = TRUE,
add_query = FALSE,
add_query_hide_credentials = TRUE
)

Arguments

location_unit_type
Either "coordinates" (for buffer around single point) or type of geographic location, including: "countries", "regions", "cities", "zips", "places", "geo_markets", "electoral_district", or "country_groups". See the Basic Targetting documentation for more information.

If location_unit_type is "coordinates"

lat_lon Coordinates, c(lat, lon). For example, c(38.90, -77.01)
radius Radius around coordinate
radius_unit Unit for radius; either "kilometer" or "mile"

If location_unit_type is not "coordinates"

location_keys Key associated with location. Use the get_fb_parameter_ids function to get location keys; see here for examples.

Other location parameters
location_types  Either: (1) "home" (people whose stated location in Facebook profile "current city" is in an area, validated by IP), (2) "recent" (people whose recent location is in the selected area, determined by mobile device data), (3) "travel_in" (people whose most recent location is in selected area and more than 100 miles from stated current city), (4) c("home", "recent") (for people in either category)

locales  Locales ID. For more information on locales, see the Advanced Targeting Documentation

---

Parameters:
- Within parameters, vectors (c()) specify OR conditions and lists (list()) specify AND conditions. For example, interests = c(6003349442621, 6003139266461) will target users who are interested in either entertainment OR movies, while interests = list(6003349442621, 6003139266461) will target users who are interested in either entertainment AND movies.
- Across parameters, AND conditions are used. For example, if enter interests = 6003349442621 and behaviors = 6008297697383 are specified, the function will query Facebook users interested in entertainment AND are frequent travelers.
- Or conditions across parameters can be specified using the flex_target argument; see package documentation for examples.

interests  Interest IDs. For example, interests = c(6003349442621, 6003139266461) will target users who are interested in either entertainment or movies. Use get_fb_parameter_ids(type = "interests", ...) to get dataframe with IDs and descriptions. For more information, see the Basic Targeting Documentation.

behaviors  Behavior IDs. For example, behaviors = c(6002714895372, 6008297697383) will target users who are either frequent travelers or returned from travels 2 weeks ago. Use get_fb_parameter_ids(type = "behaviors", ...) to get dataframe with IDs and descriptions. For more information, see the Basic Targeting Documentation.

college_years  College graduation years. For example, college_years = c(2014, 2015) will target users who graduated college in 2014 or 2015. For more information, see the Advanced Targeting Documentation.

education_majors  Education major IDs. For example, education_majors = 123 will target users who majored in computer science. Use get_fb_parameter_ids(type = "education_majors", q = "Computer", ...) to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

education_schools  School IDs. For example, education_schools = 105930651606 will target users at/who graduated from Harvard University. Use get_fb_parameter_ids(type = "education_schools", q = "Harvard", ...) to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

education_statuses  Education status IDs. For example, education_statuses = c(9,10) will yield those who report to have either a Master degree or professional degree. Use
get_fb_parameter_ids(type = "education_statuses", ...) to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

family_statuses
Family status IDs. For example, family_statuses = c(6023080302983, 6023005681983) targets users who are parents with preteens or parents with teenagers. Use get_fb_parameter_ids(type = "family_statuses") to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

income
Income IDs. For example, income = c(6107813553183, 6107813554583) targets users with a household income in the top 10%-25% or 25%-50% of ZIP codes (US). Use get_fb_parameter_ids(type = "income") to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

industries
Industries IDs. For example, industries = c(6008888980183, 6008888972183) targets users who work in sales or legal services. Use get_fb_parameter_ids(type = "industries") to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

life_events
Life event IDs. For example, life_events = c(6005149512172, 6005149512172) targets users who recently moved or are in a new job. Use get_fb_parameter_ids(type = "life_events") to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

relationship_statuses
Relation status IDs. For example, relationship_statuses = c(3,4) targets those who are married or engaged. Use get_fb_parameter_ids(type = "relationship_statuses") to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

work_positions
Work position IDs. For example, work_positions = 105763692790962 will target users who indicate they are contractors. Use get_fb_parameter_ids(type = "work_positions", ...) to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

work_employers
Work employer IDs. For example, work_employers = 50431654 will target users who work for Microsoft. Use get_fb_parameter_ids(type = "work_employers", ...) to get dataframe with IDs and descriptions. For more information, see the Advanced Targeting Documentation.

excl_interests
Interest IDs to exclude.

excl_behaviors
Behavior IDs to exclude.

excl_college_years
College year IDs to exclude.

excl_education_majors
Education major IDs to exclude.

excl_education_schools
Education school IDs to exclude.
excl_education_statuses
   Education status IDs to exclude.
excl_family_statuses
   Family status IDs to exclude.
excl_income
   Income IDs to exclude.
excl_industries
   Industry IDs to exclude.
excl_life_events
   Life event IDs to exclude.
excl_relationship_statuses
   Relationship status IDs to exclude.
excl_work_positions
   Work position IDs to exclude.
excl_work_employers
   Work employer IDs to exclude.

-----------------------------:

Non-Flexible parameters:

• Across parameters, AND conditions are used. For example, if gender = 1 and age_min = 30, queries users that are male AND are over 30 years old.
• These parameters cannot be specified in flex_targeting
• Within parameters, vectors (c()) specify OR conditions. AND conditions cannot be specified within these parameters.

user_os
   User operating systems. For example, user_os = ('iOS', 'Android') targets those that use either an iOS or Android OS; user_os = c("Android_ver_4.2_and_above") targets those using Android version 4.2 and above; and user_os = c("iOS_ver_8.0_to_9.0") targets those using iOS version 8.0 to 9.0. Different versions can be specified. For more information, see the Advanced Targeting Documentation.
wireless_carrier
   Wireless carrier. If set to Wifi, then targets those connecting via a Wifi network. For more information, see the Advanced Targeting Documentation.
gender
   Genders to target; 1 targets males and 2 targets females. Default is both. See gender in the Basic Targeting Documentation.
age_min
   Minimum age. Default is 18. See age_min in the Basic Targeting Documentation.
age_max
   Maximum age. Default is 65. See age_max in the Basic Targeting Documentation.

-----------------------------:

Flex targeting parameters

flex_target
   Flexible targeting allows for more complicated parameter specifications. For example, specifying OR conditions across parameter types (eg, behaviors, interests, etc). For information on how to use flexible targeting, see the documentation here.

-----------------------------:

Parameters for credentials
query_fb_marketing_api

version  API version. e.g., "v19.0"
creation_act  Facebook creation act
token  Facebook API token

Scraping parameters

sleep_time  How much time (in seconds) to pause between each query (default: 0.1).
show_result  After each query, whether to print the number of monthly active users (default: FALSE).
verbose  Display messages that indicate the function is pausing before making additional queries. Pausing can result from API key rate limits or no internet (default: TRUE).

Return query text

add_query  If TRUE, add query text as variable in returned dataframe
add_query_hide_credentials  If TRUE (and add_query is TRUE), hide the creation_act and token from the query text returned in the dataframe

Value

Dataframe that includes (1) daily and monthly active users and (2) parameter values

See Also

get_fb_parameter_ids() To get IDs and descriptions for behaviors, demographics, interests, and locales. For additional details on how to use the package, see the documentation here.

Examples

```r
## Not run:
#### Define version, creation act, and token
VERSION = "enter-version"
CREATION_ACT = "creation_act"
TOKEN = "enter-token"

#### Query data
## All Facebook users in US
query_fb_marketing_api(
  location_unit_type = "countries",
  location_keys = "US",
  version = VERSION,
  creation_act = CREATION_ACT,
  token = TOKEN)

## All Facebook users in US with interest in concernts
concert_id <- get_fb_parameter_ids(type = "interests", version = VERSION, token = TOKEN) %>%
  filter(name == "Concerts (music event)") %>%
pull(id)
```
query_fb_marketing_api(
    location_unit_type = "countries",
    location_keys = "US",
    interests = concert_id,
    version = VERSION,
    creation_act = CREATION_ACT,
    token = TOKEN)

## End(Not run)
Index

get_fb_parameter_ids, 2
get_fb_parameter_ids(), 13
get_fb_suggested_radius, 4
get_location_coords, 5

map_param, 6
map_param_vec, 7

query_fb_marketing_api, 8