

# Package: rsocialwatcher (via r-universe)

September 3, 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/>](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](https://doi.org/10.1038/s41598-023-49564-6)), disease surveillance (Araujo et al. (2017) [doi:10.48550/arXiv.1705.04045](https://doi.org/10.48550/arXiv.1705.04045)), and measuring migration (Alexander, Polimis, and Zagheni (2020) [doi:10.1007/s11113-020-09599-3](https://doi.org/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

**Roxygen** list(markdown = TRUE)

**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

**Repository** <https://worldbank.r-universe.dev>

**RemoteUrl** <https://github.com/worldbank/rsocialwatcher>

**RemoteRef** HEAD

**RemoteSha** d9818ed6b1153aa078eafaf3ef533bf63b7a5e8d

## Contents

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

<b>Index</b>	<b>15</b>
--------------	-----------

---

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

```
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	Type of data. Either: "behaviors", "demographics", "interests", "income", "industries", "life_events", "family_statuses", "work_positions", "work_employers", "education_statuses", "relationship_statuses", "education_majors", "locales", "country", "country_group", "region", "large_geo_area", "medium_geo_area", "small_geo_area", "city", "subcity", "neighborhood", "zip", "geo_market", "electoral_district", "zip"
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."

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 (eg, 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

```
## 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,
```

```
token      = TOKEN)

## End(Not run)
```

---

```
get_fb_suggested_radius
  Get Suggested Radius
```

---

### Description

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

### Usage

```
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

```
## 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

```
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 <b>Basic Targetting</b> documentation for more information.
location_keys	Key associated with location. Use the <code>get_fb_parameter_ids</code> function to get location keys; see <a href="#">here</a> 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 <code>large_query_pause</code> can be set to > 0 to sleep for <code>large_query_pause</code> 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**

```
## 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)
```

---

map_param	<i>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.</i>
-----------	---

---

**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**

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

**Arguments**

```
...           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**

```

## 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 (ie, 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 (ie, 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)
```

---

```
query_fb_marketing_api
```

*Query 'Facebook Marketing' API*

---

**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,
    creation_act,
    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 <a href="#">Advanced Targeting Documentation</a>
<hr/> <p><b>Parameters:</b></p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• Or conditions across parameters can be specified using the flex_target argument; see <a href="#">package documentation</a> for examples.</li> </ul>	
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 <a href="#">Basic Targeting Documentation</a> .
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 <a href="#">Basic Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
relationship_statuses	Relationship 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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .
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 <a href="#">Advanced Targeting Documentation</a> .

---

#### Exclude Facebook users from query by select parameters

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.
<hr/>	
<b>Non-Flexible parameters:</b>	
	<ul style="list-style-type: none"> <li>• 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.</li> <li>• These parameters <i>cannot</i> be specified in flex_targeting</li> <li>• Within parameters, vectors (c()) specify OR conditions. AND conditions cannot be specified within these parameters.</li> </ul>
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 <a href="#">Advanced Targeting Documentation</a> .
wireless_carrier	Wireless carrier. If set to Wifi, then targets those connecting via a Wifi network. For more information, see the <a href="#">Advanced Targeting Documentation</a> .
gender	Genders to target; 1 targets males and 2 targets females. Default is both. See gender in the <a href="#">Basic Targeting Documentation</a> .
age_min	Minimum age. Default is 18. See age_min in the <a href="#">Basic Targeting Documentation</a> .
age_max	Maximum age. Default is 65. See age_max in the <a href="#">Basic Targeting Documentation</a> .
<hr/>	
<b>Flex targeting parameters</b>	
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 <a href="#">the documentation here</a> .
<hr/>	
<b>Parameters for credentials</b>	

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

```
## 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 concerts
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