Package 'togglr'

October 15, 2026

Type Package

Title Client for the 'Toggl Track' API

Version 1.0.0

Description Read and write time entries, projects and clients of the 'Toggl Track' time tracker <https://toggl.com> from R, and download its reports. Reads never return silently truncated or empty results: HTTP errors are raised, the hourly quota and rate limits are respected, and paginated reports are checked for completeness.

License GPL-3

Depends R (>= 4.1.0)

URL https://github.com/ThinkR-open/togglr

BugReports https://github.com/ThinkR-open/togglr/issues

Imports assertthat, cachem (>= 1.0.0), dplyr (>= 1.0.0), filelock (>= 1.0.2), getPass, glue, httr (>= 1.4.0), jsonlite (>= 1.7.0), keyring, lubridate (>= 1.7.0), openssl, magrittr, parsedate, prettyunits, purrr, rappdirs, rstudioapi, stats, utils

Suggests callr, covr, knitr, pkgload, rmarkdown, testthat (>= 3.1.7), withr (>= 2.5.0)

VignetteBuilder knitr

ByteCompile true

Config/testthat/edition 3

Encoding UTF-8

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Vincent Guyader [aut, cre], ThinkR [cph, fnd]

Maintainer Vincent Guyader <vincent@thinkr.fr>

Repository CRAN

Date/Publication 2026-10-15 08:50:02 UTC

Contents

togglr-package . . . 3
ask_toggl_api_token . . . 4
clean_cache . . . 4
clean_memoise_cache . . . 5
client_id_to_name . . . 5
client_name_to_id . . . 6
create_client . . . 7
delete_toggl_api_token . . . 7
get_all_client_info . . . 8
get_all_client_names . . . 8
get_all_project_names . . . 9
get_client_project . . . 10
get_context . . . 10
get_context_project . . . 11
get_current . . . 11
get_current_duration . . . 12
get_current_project_task_detail . . . 12
get_dashboard . . . 13
get_detailled_report . . . 14
get_project_data . . . 15
get_project_id . . . 16
get_project_id_and_name . . . 17
get_project_total . . . 17
get_summary_report . . . 18
get_time_entries . . . 19
get_toggl_api_token . . . 20
get_weekly_report . . . 20
get_workspaces . . . 21
get_workspace_id . . . 21
get_workspace_users . . . 22
notification . . . 22
open_toggl_website_app . . . 23
open_toggl_website_profile . . . 23
set_toggl_api_token . . . 24
simplify . . . 24
toggl_create . . . 25
toggl_create_project . . . 26
toggl_start . . . 27
toggl_stop . . . 28
toggl_update_entries . . . 28
to_humain . . . 30
update_toggl_api_token . . . 30

Index **31**

`togglr-package` *togglr: client for the 'Toggl Track' API*

Description

Track time spent in R with the toggl.com timesheet: read and write time entries, projects and clients, and download reports (Reports API v3).

Options

Every request goes through the same policy (quota, rate limits, retries), tuned by these options:

`togglr.min_interval` minimal number of seconds between two requests (1). Raise it (e.g. 16) to spread a long extraction over the hourly quota of the API.

`togglr.max_wait` maximal number of seconds a call waits for a retry or for the reset of an exhausted quota (60); beyond, it fails without sending the request.

`togglr.timeout` timeout of a request, in seconds (120).

`togglr.state_file` path of a file sharing the pacing of the requests and the known quota between R processes (callr, future, targets workers, cron jobs), under a lock. Without it, this state is private to the session.

`togglr.quota_spread` when an answer announces fewer remaining requests than this (10), the next ones are spread until the reset of the quota.

`togglr.page_size` number of entries per page of the detailled report (1000). With `togglr.max_pages`, it caps the entries read in one period: 100 000 by default.

`togglr.report_attempts` number of reads of a period that keeps changing during its pagination (3), before an error.

`togglr.max_pages` maximal number of pages read by a paginated call (100), against an endless pagination.

`togglr.cache_dir` folder of the reports cache (`get_detailled_report(cache = TRUE)`).

Quota

Toggl counts an hourly quota per organization (endpoints of a workspace, reports included) and per user (`/me`). togglr remembers an exhausted quota per token and per scope; two workspaces of one organization each learn it from their own 402 answer.

Author(s)

vincent vincent@thinkr.fr

See Also

Useful links:

- https://github.com/ThinkR-open/togglr
- Report bugs at https://github.com/ThinkR-open/togglr/issues

ask_toggl_api_token *ask_toggl_api_token*

Description

ask for the toggle api token

Usage

```
ask_toggl_api_token(msg = "toggl api token")
```

Arguments

msg	the message

Value

the token typed by the user, or NULL when nothing was typed

clean_cache *Empty the cache of the reports*

Description

Removes the periods stored by `get_detailled_report(cache = TRUE)`.

Usage

```
clean_cache(cache_dir = toggl_cache_dir())
```

Arguments

cache_dir	folder of the cache

Value

invisibly, the result of `unlink()` (0 for success)

Examples

```
## Not run:
clean_cache()

## End(Not run)
```

clean_memoise_cache *clean memoise cache (deprecated)*

Description

Deprecated: togglr no longer memoises. Removes the cache folder of togglr 0.2.x (and the reports cache it contains). Use `clean_cache()`.

Usage

```
clean_memoise_cache(
  memoise_cache_dir = getOption("togglr_memoise_dir", default =
    rappdirs::user_cache_dir("togglr"))
)
```

Arguments

`memoise_cache_dir`	folder to remove

Value

invisibly, the result of `unlink()` (0 for success)

Examples

```
## Not run:
clean_memoise_cache()

## End(Not run)
```

client_id_to_name *client_id_to_name*

Description

return client name from client id

Usage

```
client_id_to_name(
  id,
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

id	client id
api_token	the toggl api token
workspace_id	workspace id

Value

the client name

`client_name_to_id` *client_name_to_id*

Description

return client id from client name

Usage

```
client_name_to_id(
  name,
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

name	client name
api_token	the toggl api token
workspace_id	workspace id

Value

the client id

create_client *create_client*

Description

create a client

Usage

```
create_client(
  name = "wihtout client",
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

name	client name
api_token	the toggl api token
workspace_id	workspace id

Value

the created client, as returned by the API (a list with its id)

Examples

```
## Not run:
get_toggl_api_token()# set your api token here
create_client("new client")

## End(Not run)
```

delete_toggl_api_token *delete_toggl_api_token*

Description

delete the toggle api token

Usage

```
delete_toggl_api_token()
```

Value

invisibly, NULL or the error caught when there was no key

get_all_client_info *get_all_client_info*

Description

get_all_client_info

Usage

```
get_all_client_info(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

api_token	the toggl api token
workspace_id	workspace_id

Value

a data.frame

Examples

```
## Not run:
get_all_client_info()

## End(Not run)
```

get_all_client_names *get all client's name*

Description

get all client's name

Usage

```
get_all_client_names(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  since = toggl_today() %m-% lubridate::years(1),
  until = toggl_today()
)
```

Arguments

api_token	the toggl api token
workspace_id	the workspace id
since	a date
until	a date

Value

a character vector of the client names, without duplicates

---

get_all_project_names *Get all project's names*

Description

Get all project's names

Usage

```
get_all_project_names(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  since = toggl_today() %m-% lubridate::years(1),
  until = toggl_today(),
  user_ids = NULL
)
```

Arguments

api_token	the toggl api token
workspace_id	the workspace id
since	a date
until	a date
user_ids	optional, only the projects these users worked on

Value

a character vector of the project names

get_client_project *get_client_project*

Description

get_client_project

Usage

```
get_client_project(
  id,
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

id	client id
api_token	the toggl api token
workspace_id	workspace_id

Value

the client name

get_context *get_context*

Description

retrieve Rstudio projet if possible

Usage

```
get_context()
```

Value

a text describing the RStudio project, or "I'm using R" outside a project

get_context_project *get_context_projet*

Description

retrieve Rstudio projet if possible

Usage

```
get_context_project()
```

Value

the name of the RStudio project, or "sans projet" outside a project

get_current *get_current*

Description

retrieve current projet id

Usage

```
get_current(api_token = get_toggl_api_token())
```

Arguments

api_token	the toggl api token

Value

a list describing the running time entry (`id`, `description`, `start`, `project_id`...), or `NULL` when nothing is running

get_current_duration *get current duration*

Description

get current duration

Usage

```
get_current_duration(api_token = get_toggl_api_token())
```

Arguments

api_token	the Token API

Value

a difftime

Examples

```
## Not run:
get_current_duration()

## End(Not run)
```

get_current_project_task_detail *get all task from a project*

Description

get all task from a project

Usage

```
get_current_project_task_detail(
  project_name = get_context_project(),
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  since = toggl_today() %m-% lubridate::years(1),
  until = toggl_today(),
  humain = TRUE
)
```

Arguments

project_name	project name
api_token	the toggl api token
workspace_id	the workspace id
since	a date
until	a date
humain	boolean humain readable time

Value

a data frame of the tasks of the project (title, time...), the running entry included

---

get_dashboard *get all data from* https://toggl.com

Description

get all data from https://toggl.com

Usage

```
get_dashboard(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  since = toggl_today() %m-% lubridate::years(1),
  until = toggl_today(),
  user_ids = NULL
)
```

Arguments

api_token	the toggl api token
workspace_id	the workspace id
since	begin date
until	stop date
user_ids	optional, ids of the users whose time is counted (all by default)

Value

an object of class toggl: a list with synthese (one row per project: id, project, client, time in ms) and tache (per project, a data frame of its time entries: title, time in ms, local_start)

`get_detailled_report` *Detailled report: every time entry of a period*

Description

Downloads the time entries of the workspace between `since` and `until` (both included) from the reports API v3, all pages included.

Usage

```
get_detailled_report(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  since = toggl_today() %m-% lubridate::years(1),
  until = toggl_today(),
  user_agent = toggl_user_agent(),
  max_page = NULL,
  cache = FALSE,
  cache_dir = toggl_cache_dir(),
  memoise_cache_dir = NULL,
  user_ids = NULL
)
```

Arguments

`api_token`	the toggl api token
`workspace_id`	the workspace id
`since`	begin date (included): a `Date`, a date-time or a text "YYYY-MM-DD"; one year ago by default
`until`	end date (included), same formats; today by default
`user_agent`	ignored since the move to Reports v3 (the user agent is sent in the header of every request)
`max_page`	deprecated, ignored: every page is read.
`cache`	logical. If `TRUE`, a period fully fetched and checked is stored on disk and returned as is by later calls with the same `since` / `until`. Only use it for closed periods: entries created or edited later in Toggl will not be seen. Failures are never cached. Default `FALSE`.
`cache_dir`	folder of the cache used when `cache = TRUE`: by default `options("togglr.cache_dir")`, or the `reports` sub-folder of `rappdirs::user_cache_dir("togglr")`
`memoise_cache_dir`	deprecated, use `cache_dir`
`user_ids`	optional, ids of the users whose entries are wanted (all the users of the workspace by default)

Details

Columns: id, pid, tid, uid, description, start, end, updated (text, ISO 8601), dur (milliseconds), user, use_stop, client, project, project_color, project_hex_color, task, billable, is_billable, cur, tags (list), then any column added by the API.

The entries come from Reports v3 (/search/time_entries, enriched with the names of users, projects, clients and tags), with large pages (options(togglr.page_size), 1000 by default). The period is checked against the total duration announced by the API before and after the pagination, and against the edit time of every entry received; if entries were added, removed or edited meanwhile, the period is fetched again, up to options(togglr.report_attempts) reads in all (3 by default), then an error is raised. One movement escapes these checks: an entry deleted and another one of exactly the same duration created during the read, when the new entry is not in the pages received. The rounding set in the Toggl profile is not applied. A running entry is left out. An HTTP error (quota, server error, authentication) raises an error too: the function never returns a silently truncated period.

Periods longer than a year (365 days or more between since and until, across a new year) are split by calendar year.

Value

a tibble, one row per time entry, with fixed column types (see Details); 0 rows for an empty period

---

get_project_data *Data of one project*

Description

Data of one project

Usage

```
get_project_data(
  project_id,
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

project_id	the project id
api_token	the toggl api token
workspace_id	the workspace id

Value

a list with the fields of the project (id, name, client_id, active, billable, color...), as returned by the API v9

Examples

```
## Not run:
get_project_data(get_project_id("my_project"))

## End(Not run)
```

`get_project_id` *get_project_id*

Description

retrieve project id

Usage

```
get_project_id(
  project_name = get_context_project(),
  api_token = get_toggl_api_token(),
  create = FALSE,
  client = NULL,
  workspace_id = get_workspace_id(api_token),
  color = NULL
)
```

Arguments

`project_name`	the project name
`api_token`	the toggl api token
`create`	boolean do you want to create the project if it doesnt exist ?
`client`	client name
`workspace_id`	workspace id
`color`	id of the color selected for the project

Value

the id of the project, or `NULL` when it does not exist (and is not created)

get_project_id_and_name *get_project_id_and_name*

Description

retrieve project id and name

Usage

```
get_project_id_and_name(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  active = NULL
)
```

Arguments

api_token	the toggl api token
workspace_id	workspace id
active	NULL for the active projects only (the default of the API), "false" for the archived ones, "both" for all

Value

a tibble with id, project_name and active; 0 rows when the workspace has no such project

get_project_total *get project total time*

Description

get project total time

Usage

```
get_project_total(
  project_name = get_context_project(),
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  since = toggl_today() %m-% lubridate::years(1),
  until = toggl_today()
)
```

Arguments

project_name	project name
api_token	the toggl api token
workspace_id	the workspace id
since	a date
until	a date

Details

The rounding set in the Toggl profile is not applied (rounding = 0 in the request), as in Reports v2.

Value

the total time spent on the project, as text (e.g. "12:30:00"), the running entry included

---

get_summary_report *get summary report*

Description

by user then projet

Usage

```
get_summary_report(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  since = toggl_today() %m-% lubridate::years(1),
  until = toggl_today(),
  user_agent = toggl_user_agent(),
  users = get_workspace_users(api_token = api_token, workspace_id = workspace_id)
)
```

Arguments

api_token	the toggl api token
workspace_id	the workspace id
since	begin date
until	stop date
user_agent	ignored since the move to Reports v3 (the user agent is sent in the header of every request)
users	the users of the workspace (a data frame with id and fullname), to name the rows

Details

The rounding set in the Toggl profile is not applied (`rounding = 0` in the request), as in Reports v2.

Value

a tibble, one row per user: `id`, `user` (full name), `time` (ms) and `items` (a data frame of the user's time entries: `title`, `time` in ms, `local_start`; `cur`, `sum` and `rate` are `NA` since Reports v3)

`get_time_entries` *Get all time entries between 2 dates*

Description

Get all time entries between 2 dates

Usage

```
get_time_entries(
  api_token = get_toggl_api_token(),
  since = Sys.time() - lubridate::weeks(1),
  until = Sys.time(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

`api_token`	the toggl api token
`since`	begin date (One week ago by default)
`until`	stop date (Now by defaut)
`workspace_id`	the workspace id (to name the projects and clients)

Value

a data.frame containing all time entries, with `project_name` and `client_name`. The endpoint goes back at most 3 months: for an older period, use `get_detailled_report()` with `user_ids`.

Examples

```
## Not run:
get_time_entries()

## End(Not run)
```

get_toggl_api_token *get_toggl_api_token*

Description

return the toggle api token

Usage

```
get_toggl_api_token(ask = TRUE)
```

Arguments

ask	booleen do we have to ask if missing

Value

the token (a character string), or NULL when there is none and ask = FALSE

get_weekly_report *get weekly report*

Description

get weekly report

Usage

```
get_weekly_report(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  since = toggl_today() %m-% lubridate::years(1),
  until = toggl_today(),
  user_agent = toggl_user_agent()
)
```

Arguments

api_token	the toggl api token
workspace_id	the workspace id
since	begin date
until	stop date
user_agent	ignored since the move to Reports v3 (the user agent is sent in the header of every request)

Details

The rounding set in the Toggl profile is not applied (`rounding = 0` in the request), as in Reports v2.

Value

a tibble, one row per project: `pid`, `title` (`project`, `client`, `color`, `hex_color`; Reports v3 gives no color index, `color` is always `"0"`), `totals` (8 values in ms: the 7 days, then the week; `NA` for 0) and `details` (the same per user); 0 rows for a week without any entry

---

`get_workspaces` *List the workspaces of the account*

Description

List the workspaces of the account

Usage

```
get_workspaces(api_token = get_toggl_api_token())
```

Arguments

`api_token`	the toggl api token

Value

a tibble with `id`, `name` and `default` (`TRUE` for the default workspace of the account)

`get_workspace_id` *get_workspace_id*

Description

Return the id of the default workspace of the account (as set in Toggl), or of its first workspace when no default is set.

Usage

```
get_workspace_id(api_token = get_toggl_api_token())
```

Arguments

`api_token`	the toggl api token

Value

the workspace id (an integer)

See Also

`get_workspaces()` to list every workspace

`get_workspace_users` *Users of a workspace*

Description

Users of a workspace

Usage

```
get_workspace_users(
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

`api_token`	the toggl api token
`workspace_id`	the workspace id

Value

a tibble, one row per user of the workspace (`id`, `fullname`, `email`...)

`notification` *show notification*

Description

show notification using message

Usage

```
notification(title, msg)
```

Arguments

`title`	title of the notificatin
`msg`	content of the notification

Value

invisibly, `NULL`; called for its message

`open_toggl_website_app` *open toggle.com website on timesheet page*

Description

open toggle.com website on timesheet page

Usage

```
open_toggl_website_app()
```

Value

invisibly, the result of `utils::browseURL()`

`open_toggl_website_profile` *open toggle.com website on profile page*

Description

open toggle.com website on profile page

Usage

```
open_toggl_website_profile()
```

Value

invisibly, the result of `utils::browseURL()`

set_toggl_api_token *set_toggl_api_token*

Description

set the toggle api token

Usage

```
set_toggl_api_token(token)
```

Arguments

token	toggl api token

Value

invisibly, the token stored (or NULL when nothing was typed)

simplify *Simplify a parsed JSON answer into vectors and data frames*

Description

Same result as `jsonlite::fromJSON(simplifyVector = TRUE)` on a list already parsed with `simplifyVector = FALSE` (what `httr::content()` returns), using only the public API of jsonlite: the list is serialised back to JSON at full precision, then read with simplification.

Usage

```
simplify(x, ...)
```

Arguments

x	a list, as returned by `httr::content()`
...	passed to `jsonlite::fromJSON()` (e.g. `simplifyDataFrame`)

Value

the simplified object

toggl_create *toggl_create*

Description

create a time entry

Usage

```
toggl_create(
  description = get_context(),
  start = now(),
  pid = get_project_id(api_token = api_token, workspace_id = workspace_id),
  stop,
  duration,
  tags = NULL,
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

description	the task you did
start	time in POSIXt
pid	pid
stop	time in POSIXt
duration	in seconds
tags	tags
api_token	the toggl api token
workspace_id	workspace id

Value

invisibly, the id of the created time entry (NA, with a warning, when the API accepted it but its answer could not be read)

Examples

```
## Not run:
set_toggl_api_token() # once, stores your token in the keyring
toggl_create(duration=1200)

toggl_create( description="description",
             start=now(),
             pid = get_project_id(project_name = "projectname",
                                  create=TRUE,client = "client"),
```

```
                duration=1000,
                api_token=get_toggl_api_token())




## End(Not run)
```

`toggl_create_project` *toggl_create_project*

Description

create a project

Usage

```
toggl_create_project(
  project_name = get_context_project(),
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token),
  client = NULL,
  private = FALSE,
  color = NULL,
  active = TRUE
)
```

Arguments

`project_name`	project name
`api_token`	the toggl api token
`workspace_id`	workspace_id
`client`	client name: an existing client is reused, an unknown one is created; `NULL` or `""` for no client
`private`	whether project is accessible for only project users or for all workspace users (boolean, default false)
`color`	color of the project, as a hex code (e.g. "#06aaf5")
`active`	boolean set project as active, TRUE by default

Value

the id of the project (created, or already existing)

Examples

```
## Not run:
toggl_create_project()

## End(Not run)
```

---

toggl_start *toggl_start*

Description

start a task

Usage

```
toggl_start(
  description = get_context(),
  client = "without client",
  project_name = get_context_project(),
  start = now(),
  api_token = get_toggl_api_token(),
  tags = "",
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

description	the task you are doing
client	client name
project_name	nom du projet
start	start time in POSIXt
api_token	the toggl api token
tags	tags
workspace_id	workspace id

Value

invisibly, the id of the started time entry (NA, with a warning, when the API accepted it but its answer could not be read)

Examples

```
## Not run:
get_toggl_api_token()# set your api token here
toggl_start()

## End(Not run)
```

toggl_stop *toggl_stop*

Description

stop the active task

Usage

```
toggl_stop(
  current = get_current(api_token = api_token),
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

current	list task id and start time
api_token	the toggl api token
workspace_id	workspace id

Value

invisibly, the stopped time entry, as returned by the API

Examples

```
## Not run:
set_toggl_api_token() # once, stores your token in the keyring
toggl_start()
toggl_stop()

## End(Not run)
```

toggl_update_entries *toggl_update_entries*

Description

update time entries

Usage

```
toggl_update_entries(
  time_entry_ids,
  description = NULL,
  start = NULL,
  pid = NULL,
  stop = NULL,
  duration = NULL,
  tags = NULL,
  api_token = get_toggl_api_token(),
  workspace_id = get_workspace_id(api_token)
)
```

Arguments

time_entry_ids	id(s) of time entries to update: one PUT request is sent per entry
description	the task you did
start	time in POSIXt
pid	project id
stop	time in POSIXt
duration	in seconds
tags	tags
api_token	the toggl api token
workspace_id	workspace id

Value

the updated entry, as returned by the API, or a list of them when several ids are given

Examples

```
## Not run:
set_toggl_api_token() # once, stores your token in the keyring
time_entry_id <- toggl_create(duration=1200)

toggl_update_entries( time_entry_id,
                    description = "new description",
                    duration = 100,
                    tags = c("tag1", "tag2"),
                    api_token=get_toggl_api_token())


## End(Not run)
```

to_humain *transforme time column into humain readable column*

Description

transforme time column into humain readable column

Usage

```
to_humain(df)
```

Arguments

df	a data.frame

Value

the data frame, with `time` as a readable duration

update_toggl_api_token *update_toggl_api_token*

Description

update the toggle api token

Usage

```
update_toggl_api_token()
```

Value

invisibly, the new token, or `NULL` when the input was cancelled (the stored token is then kept)

Index

ask_toggl_api_token, 4

clean_cache, 4
clean_cache(), *5*
clean_memoise_cache, 5
client_id_to_name, 5
client_name_to_id, 6
create_client, 7

delete_toggl_api_token, 7

get_all_client_info, 8
get_all_client_names, 8
get_all_project_names, 9
get_client_project, 10
get_context, 10
get_context_project, 11
get_current, 11
get_current_duration, 12
get_current_project_task_detail, 12
get_dashboard, 13
get_detailled_report, 14
get_detailled_report(), *19*
get_project_data, 15
get_project_id, 16
get_project_id_and_name, 17
get_project_total, 17
get_summary_report, 18
get_time_entries, 19
get_toggl_api_token, 20
get_weekly_report, 20
get_workspace_id, 21
get_workspace_users, 22
get_workspaces, 21
get_workspaces(), *22*

notification, 22

open_toggl_website_app, 23
open_toggl_website_profile, 23

set_toggl_api_token, 24
simplify, 24

to_humain, 30
toggl_create, 25
toggl_create_project, 26
toggl_start, 27
toggl_stop, 28
toggl_update_entries, 28
togglr *(*togglr-package*)*, 3
togglr-package, 3

unlink(), *4, 5*
update_toggl_api_token, 30
utils::browseURL(), *23*