

Package ‘rtodoist’

February 14, 2026

Title Create and Manage Todolist using 'Todoist.com' API

Version 0.4.0

Description Allows you to interact with the API of the ``Todoist'' platform. 'Todoist' <<https://www.todoist.com/>> provides an online task manager service for teams.

License MIT + file LICENSE

URL <https://github.com/ThinkR-open/rtodoist>

BugReports <https://github.com/ThinkR-open/rtodoist/issues>

Depends R (>= 3.5.0)

Imports curl, digest, dplyr, getPass, glue, httr2, keyring, magrittr, purrr, stats, stringi, stringr, utils

Suggests covr, jsonlite, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Cervan Girard [aut] (ORCID: <<https://orcid.org/0000-0002-4816-4624>>, note: previous maintainer),
Vincent Guyader [cre, aut] (ORCID: <<https://orcid.org/0000-0003-0671-9270>>),
ThinkR [cph, fnd]

Maintainer Vincent Guyader <vincent@thinkr.fr>

Repository CRAN

Date/Publication 2026-02-14 14:40:15 UTC

Contents

accept_invitation	4
add_comment	4
add_filter	5
add_label	6
add_project	7
add_reminder	8
add_responsible_to_task	9
add_section	9
add_tasks_in_project	10
add_tasks_in_project_from_df	11
add_users_in_project	12
add_user_in_project	13
archive_project	14
archive_section	15
ask_todoist_api_token	15
call_api	16
call_api_rest	16
close_task	17
delete_collaborator	17
delete_comment	18
delete_filter	19
delete_invitation	19
delete_label	20
delete_project	21
delete_reminder	22
delete_section	22
delete_task	23
delete_todoist_api_token	24
delete_upload	24
download_backup	25
escape_json	25
export_template	26
get_activity_logs	27
get_all_data	28
get_all_filters	28
get_all_labels	29
get_all_projects	30
get_all_reminders	30
get_all_sections	31
get_all_users	31
get_all_workspaces	32
get_archived_projects	33
get_backups	33
get_comment	34
get_comments	34
get_completed_tasks	35

get_filter	36
get_filter_id	36
get_label	37
get_label_id	38
get_productivity_stats	38
get_project	39
get_project_id	40
get_section	41
get_section_id	41
get_shared_labels	42
get_task	43
get_tasks	43
get_tasks_by_filter	44
get_tasks_of_project	45
get_todoist_api_token	45
get_users_id	46
get_users_in_project	46
get_user_info	47
get_workspace_users	48
import_template	48
invite_to_workspace	49
leave_workspace	50
move_section	50
move_task	51
open_todoist_website_profile	52
quick_add_task	53
random_key	53
reject_invitation	54
remove_shared_label	54
rename_shared_label	55
reopen_task	56
set_todoist_api_token	56
unarchive_project	57
unarchive_section	58
update_comment	58
update_filter	59
update_label	60
update_project	61
update_reminder	62
update_section	63
update_task	63
update_todoist_api_token	64
update_workspace	65
upload_file	66

accept_invitation *Accept a project invitation*

Description

Accept a project invitation

Usage

```
accept_invitation(  
  invitation_id,  
  invitation_secret,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

invitation_id	id of the invitation
invitation_secret	secret of the invitation
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
accept_invitation("12345", "secret123")  
  
## End(Not run)
```

add_comment *Add a comment to a task or project*

Description

Add a comment to a task or project

Usage

```
add_comment(  
    content,  
    task_id = NULL,  
    project_id = NULL,  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

content	content of the comment
task_id	id of the task (either task_id or project_id required)
project_id	id of the project (either task_id or project_id required)
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the new comment

Examples

```
## Not run:  
add_comment(content = "This is a comment", task_id = "12345")  
add_comment(content = "Project comment", project_id = "67890")  
  
## End(Not run)
```

add_filter	<i>Add a new filter</i>
------------	-------------------------

Description

Add a new filter

Usage

```
add_filter(  
    name,  
    query,  
    color = NULL,  
    is_favorite = FALSE,  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

name	name of the filter
query	filter query string (e.g., "today overdue", "p1 & #Work")
color	color of the filter
is_favorite	boolean to mark as favorite
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the new filter

Examples

```
## Not run:
add_filter("Urgent Today", query = "today & p1")
add_filter("Work Tasks", query = "#Work", color = "blue")

## End(Not run)
```

add_label

Add a new label

Description

Add a new label

Usage

```
add_label(
  name,
  color = NULL,
  is_favorite = FALSE,
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

name	name of the label
color	color of the label
is_favorite	boolean to mark as favorite
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the new label

Examples

```
## Not run:  
add_label("urgent")  
add_label("work", color = "red")  
  
## End(Not run)
```

add_project	<i>Add a new project</i>
-------------	--------------------------

Description

Add a new project

Usage

```
add_project(project_name, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

project_name	name of the new project
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the new project

Examples

```
## Not run:  
add_project("my_proj")  
  
## End(Not run)
```

`add_reminder`*Add a reminder to a task*

Description

Add a reminder to a task

Usage

```
add_reminder(  
    task_id,  
    due_date = NULL,  
    due_datetime = NULL,  
    minute_offset = NULL,  
    type = "absolute",  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

<code>task_id</code>	id of the task
<code>due_date</code>	due date for the reminder (format: YYYY-MM-DD)
<code>due_datetime</code>	due datetime for the reminder (format: YYYY-MM-DDTHH:MM:SS)
<code>minute_offset</code>	minutes before due time to remind (for relative reminders)
<code>type</code>	type of reminder: "absolute", "relative", or "location"
<code>verbose</code>	boolean that make the function verbose
<code>token</code>	todoist API token

Value

id of the new reminder

Examples

```
## Not run:  
add_reminder(task_id = "12345", due_datetime = "2024-12-25T09:00:00")  
add_reminder(task_id = "12345", minute_offset = 30, type = "relative")  
  
## End(Not run)
```

`add_responsible_to_task`*Add responsible to a task*

Description

Add responsible to a task

Usage

```
add_responsible_to_task(  
    project_id = get_project_id(project_name = project_name, token = token),  
    project_name,  
    responsible,  
    task,  
    verbose = FALSE,  
    all_users = get_all_users(token = token),  
    token = get_todoist_api_token()  
)
```

Arguments

<code>project_id</code>	id of the project
<code>project_name</code>	name of the project
<code>responsible</code>	add someone to this task with mail
<code>task</code>	the full name of the task
<code>verbose</code>	boolean that make the function verbose
<code>all_users</code>	<code>all_users</code>
<code>token</code>	todoist API token

Value

http request

`add_section`*Add section*

Description

Add section

Usage

```

add_section(
    section_name,
    project_id = get_project_id(project_name = project_name, token = token),
    project_name,
    force = FALSE,
    token = get_todoist_api_token()
)

```

Arguments

section_name	section name
project_id	id of the project
project_name	name of the project
force	boolean force section creation even if already exist
token	todoist API token

Value

section id (character)

add_tasks_in_project *Add tasks in project*

Description

Add tasks in project

Usage

```

add_tasks_in_project(
    project_id = get_project_id(project_name = project_name, token = token),
    tasks,
    project_name,
    verbose = FALSE,
    responsible = NULL,
    due = NULL,
    section_name = NULL,
    token = get_todoist_api_token(),
    all_users = get_all_users(token = token),
    update_only = FALSE,
    check_only = FALSE,
    que_si_necessaire = TRUE
)

```

Arguments

project_id	id of the project
tasks	tasks to add, as character vector
project_name	name of the project
verbose	boolean that make the function verbose
responsible	add people in project
due	due date
section_name	section name
token	todoist API token
all_users	all_users
update_only	boolean if true, only update existing (not closed) todo
check_only	check_only
que_si_necessaire	que_si_necessaire

Value

id of project (character vector)

Examples

```
## Not run:
add_project("my_proj") %>%
  add_tasks_in_project(c("First task", "Second task"))

## End(Not run)
```

```
add_tasks_in_project_from_df
  Add tasks in project
```

Description

Add tasks in project

Usage

```
add_tasks_in_project_from_df(
  project_id = get_project_id(project_name = project_name, token = token),
  tasks_as_df,
  project_name,
  verbose = FALSE,
  token = get_todoist_api_token(),
  update_only = FALSE,
```

```

    check_only = FALSE,
    que_si_necessaire = TRUE,
    all_users = get_all_users(token = token)
  )

```

Arguments

project_id	id of the project
tasks_as_df	data.frame of tasks with c("tasks_list","responsible","due","section_name") names
project_name	name of the project
verbose	boolean that make the function verbose
token	todoist API token
update_only	boolean if true only update existing (not closed) todo
check_only	boolean if true only return number of task to add
que_si_necessaire	que_si_necessaire
all_users	all_users

Value

id of project (character vector)

See Also

[add_tasks_in_project()]

add_users_in_project *Add a list of users*

Description

Add a list of users

Usage

```

add_users_in_project(
  project_id = get_project_id(project_name = project_name, token = token),
  users_email,
  project_name,
  verbose = TRUE,
  all_users = get_all_users(token = token),
  token = get_todoist_api_token()
)

```

Arguments

project_id	id of the project
users_email	emails of user as character vector
project_name	name of the project
verbose	boolean that make the function verbose
all_users	all_users
token	token

Value

id of project (character vector)

`add_user_in_project` *Add one user*

Description

Add one user

Usage

```
add_user_in_project(  
  project_id = get_project_id(project_name = project_name, token = token),  
  mail,  
  project_name,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

project_id	id of the project
mail	mail of the user
project_name	name of the project
verbose	boolean that make the function verbose
token	token

Value

id of project (character vector)

Examples

```
## Not run:
get_project_id("test") %>%
  add_user_in_project("jean@mail.fr")

## End(Not run)
```

archive_project	<i>Archive a project</i>
-----------------	--------------------------

Description

Archive a project

Usage

```
archive_project(
  project_id = get_project_id(project_name = project_name, token = token, create = FALSE),
  project_name,
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

project_id	id of the project
project_name	name of the project (for lookup if project_id not provided)
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the archived project (invisible)

Examples

```
## Not run:
archive_project(project_name = "my_proj")

## End(Not run)
```

archive_section	<i>Archive a section</i>
-----------------	--------------------------

Description

Archive a section

Usage

```
archive_section(section_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

section_id	id of the section
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the archived section (invisible)

Examples

```
## Not run:  
archive_section("12345")  
  
## End(Not run)
```

ask_todoist_api_token	<i>Pop-up to save the token</i>
-----------------------	---------------------------------

Description

Pop-up to save the token

Usage

```
ask_todoist_api_token(msg = "Register Todoist Api Token")
```

Arguments

msg	message to print in the pop-up
-----	--------------------------------

Value

password (character vector)

call_api	<i>Call the Sync API</i>
----------	--------------------------

Description

Call the Sync API

Usage

```
call_api(..., url = TODOIST_SYNC_URL, token = get_todoist_api_token())
```

Arguments

...	any params of POST request
url	url to call
token	todoist API token

Value

list

call_api_rest	<i>Call the REST API</i>
---------------	--------------------------

Description

Call the REST API

Usage

```
call_api_rest(endpoint, token = get_todoist_api_token(), ...)
```

Arguments

endpoint	API endpoint (e.g., "projects", "tasks")
token	todoist API token
...	query parameters

Value

list

close_task	<i>Close (complete) a task</i>
------------	--------------------------------

Description

Close (complete) a task

Usage

```
close_task(task_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

task_id	id of the task to close
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the closed task (invisible)

Examples

```
## Not run:  
close_task("12345")  
  
## End(Not run)
```

delete_collaborator	<i>Delete a collaborator from a project</i>
---------------------	---

Description

Delete a collaborator from a project

Usage

```
delete_collaborator(  
  project_id = get_project_id(project_name = project_name, token = token, create = FALSE),  
  project_name,  
  email,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

project_id	id of the project
project_name	name of the project (for lookup if project_id not provided)
email	email of the collaborator to remove
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the project (invisible)

Examples

```
## Not run:  
delete_collaborator(project_name = "my_proj", email = "user@example.com")  
  
## End(Not run)
```

delete_comment	<i>Delete a comment</i>
----------------	-------------------------

Description

Delete a comment

Usage

```
delete_comment(comment_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

comment_id	id of the comment
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
delete_comment("12345")  
  
## End(Not run)
```

delete_filter	<i>Delete a filter</i>
---------------	------------------------

Description

Delete a filter

Usage

```
delete_filter(  
    filter_id = get_filter_id(filter_name = filter_name, token = token),  
    filter_name,  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

filter_id	id of the filter
filter_name	name of the filter (for lookup if filter_id not provided)
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
delete_filter(filter_name = "Urgent Today")  
  
## End(Not run)
```

delete_invitation	<i>Delete an invitation</i>
-------------------	-----------------------------

Description

Delete an invitation

Usage

```
delete_invitation(
    invitation_id,
    verbose = TRUE,
    token = get_todoist_api_token()
)
```

Arguments

```
invitation_id  id of the invitation
verbose        boolean that make the function verbose
token          todoist API token
```

Value

NULL (invisible)

Examples

```
## Not run:
delete_invitation("12345")

## End(Not run)
```

delete_label	<i>Delete a label</i>
--------------	-----------------------

Description

Delete a label

Usage

```
delete_label(
    label_id = get_label_id(label_name = label_name, token = token, create = FALSE),
    label_name,
    verbose = TRUE,
    token = get_todoist_api_token()
)
```

Arguments

```
label_id      id of the label
label_name    name of the label (for lookup if label_id not provided)
verbose       boolean that make the function verbose
token         todoist API token
```

Value

NULL (invisible)

Examples

```
## Not run:  
delete_label(label_name = "urgent")  
  
## End(Not run)
```

delete_project	<i>Delete a project</i>
----------------	-------------------------

Description

Delete a project

Usage

```
delete_project(  
  project_id = get_project_id(project_name = project_name, token = token, create = FALSE),  
  project_name,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

project_id	id of the project
project_name	name of the project (for lookup if project_id not provided)
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
delete_project(project_name = "my_proj")  
  
## End(Not run)
```

delete_reminder	<i>Delete a reminder</i>
-----------------	--------------------------

Description

Delete a reminder

Usage

```
delete_reminder(reminder_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

reminder_id	id of the reminder
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
delete_reminder("12345")  
  
## End(Not run)
```

delete_section	<i>Delete a section</i>
----------------	-------------------------

Description

Delete a section

Usage

```
delete_section(section_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

section_id	id of the section
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
delete_section("12345")  
  
## End(Not run)
```

delete_task	<i>Delete a task</i>
-------------	----------------------

Description

Delete a task

Usage

```
delete_task(task_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

task_id	id of the task to delete
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
delete_task("12345")  
  
## End(Not run)
```

delete_todoist_api_token
Delete todoist api token

Description

Delete todoist api token

Usage

```
delete_todoist_api_token()
```

Value

nothing, delete the api token

delete_upload *Delete an uploaded file*

Description

Delete an uploaded file

Usage

```
delete_upload(file_url, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

file_url	URL of the file to delete
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
delete_upload("https://...")  
  
## End(Not run)
```

download_backup	<i>Download a backup</i>
-----------------	--------------------------

Description

Download a backup

Usage

```
download_backup(  
  version,  
  output_file,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

version	version of the backup to download (from get_backups())
output_file	path where to save the backup file
verbose	boolean that make the function verbose
token	todoist API token

Value

path to the saved file (invisible)

Examples

```
## Not run:  
backups <- get_backups()  
download_backup(backups$version[1], "backup.zip")  
  
## End(Not run)
```

escape_json	<i>Escape special characters for JSON</i>
-------------	---

Description

Escape special characters for JSON

Usage

```
escape_json(x)
```

Arguments

x character string to escape

Value

escaped string

export_template	<i>Export a project as a template</i>
-----------------	---------------------------------------

Description

Export a project as a template

Usage

```
export_template(
  project_id = get_project_id(project_name = project_name, token = token, create = FALSE),
  project_name,
  output_file,
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

project_id id of the project to export
 project_name name of the project (for lookup if project_id not provided)
 output_file path where to save the template file
 verbose boolean that make the function verbose
 token todoist API token

Value

path to the saved file (invisible)

Examples

```
## Not run:
export_template(project_name = "my_proj", output_file = "template.csv")

## End(Not run)
```

get_activity_logs	<i>Get activity logs</i>
-------------------	--------------------------

Description

Get activity logs

Usage

```
get_activity_logs(  
  object_type = NULL,  
  object_id = NULL,  
  event_type = NULL,  
  parent_project_id = NULL,  
  parent_item_id = NULL,  
  initiator_id = NULL,  
  since = NULL,  
  until = NULL,  
  limit = 30,  
  offset = 0,  
  token = get_todoist_api_token()  
)
```

Arguments

object_type	filter by object type (e.g., "project", "item", "note")
object_id	filter by specific object id
event_type	filter by event type (e.g., "added", "updated", "completed")
parent_project_id	filter by parent project id
parent_item_id	filter by parent item id
initiator_id	filter by user who initiated the action
since	return events since this date (format: YYYY-MM-DDTHH:MM:SS)
until	return events until this date (format: YYYY-MM-DDTHH:MM:SS)
limit	maximum number of events to return (default 30, max 100)
offset	offset for pagination
token	todoist API token

Value

tibble of activity events

Examples

```
## Not run:
get_activity_logs()
get_activity_logs(object_type = "item", event_type = "completed")

## End(Not run)
```

get_all_data	<i>Get all objects inside a list</i>
--------------	--------------------------------------

Description

Collect all the objects in a list. This allows you to explore your to-do list.

Usage

```
get_all_data(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

list of all objects

Examples

```
## Not run:
# Set API key first
set_todoist_api_token()
# Get all objects
objects <- get_all_data()

## End(Not run)
```

get_all_filters	<i>Get all filters</i>
-----------------	------------------------

Description

Get all filters

Usage

```
get_all_filters(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

tibble of all filters

Examples

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

<i>get_all_labels</i>	<i>Get all labels</i>
-----------------------	-----------------------

Description

Get all labels

Usage

```
get_all_labels(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

tibble of all labels

Examples

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

get_all_projects *List of projects*

Description

List of projects

Usage

```
get_all_projects(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

list of all projects

Examples

```
## Not run:  
# Set API key first  
set_todoist_api_token()  
# Get all projects  
projects <- get_all_projects()  
  
## End(Not run)
```

get_all_reminders *Get all reminders*

Description

Get all reminders

Usage

```
get_all_reminders(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

tibble of all reminders

Examples

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

get_all_sections	<i>Get all sections</i>
------------------	-------------------------

Description

Get all sections

Usage

```
get_all_sections(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

tibble of all sections

Examples

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

get_all_users	<i>Get users</i>
---------------	------------------

Description

Get a tibble with emails and ids of users

Usage

```
get_all_users(token = get_todoist_api_token())
```

Arguments

token token

Value

tibble of users

Examples

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

get_all_workspaces *Get all workspaces*

Description

Get all workspaces

Usage

```
get_all_workspaces(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

tibble of all workspaces

Examples

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

get_archived_projects *Get archived projects*

Description

Get archived projects

Usage

```
get_archived_projects(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

tibble of archived projects

Examples

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

get_backups *Get list of backups*

Description

Get list of backups

Usage

```
get_backups(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

tibble of available backups

Examples

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

get_comment	<i>Get a single comment by ID</i>
-------------	-----------------------------------

Description

Get a single comment by ID

Usage

```
get_comment(comment_id, token = get_todoist_api_token())
```

Arguments

comment_id	id of the comment
token	todoist API token

Value

list with comment details

Examples

```
## Not run:  
get_comment("12345")  
  
## End(Not run)
```

get_comments	<i>Get comments</i>
--------------	---------------------

Description

Get comments

Usage

```
get_comments(  
  task_id = NULL,  
  project_id = NULL,  
  token = get_todoist_api_token()  
)
```

Arguments

task_id	id of the task (either task_id or project_id required)
project_id	id of the project (either task_id or project_id required)
token	todoist API token

Value

tibble of comments

Examples

```
## Not run:
get_comments(task_id = "12345")
get_comments(project_id = "67890")

## End(Not run)
```

get_completed_tasks *Get completed tasks*

Description

Get completed tasks

Usage

```
get_completed_tasks(
  project_id = NULL,
  since = NULL,
  until = NULL,
  limit = 50,
  token = get_todoist_api_token()
)
```

Arguments

project_id	project id to filter by (optional)
since	return tasks completed since this date (optional, format: YYYY-MM-DDTHH:MM:SS)
until	return tasks completed until this date (optional, format: YYYY-MM-DDTHH:MM:SS)
limit	maximum number of tasks to return (default 50, max 200)
token	todoist API token

Value

tibble of completed tasks

Examples

```
## Not run:
get_completed_tasks()
get_completed_tasks(project_id = "12345")

## End(Not run)
```

get_filter	<i>Get a single filter by ID</i>
------------	----------------------------------

Description

Get a single filter by ID

Usage

```
get_filter(filter_id, token = get_todoist_api_token())
```

Arguments

filter_id	id of the filter
token	todoist API token

Value

list with filter details

Examples

```
## Not run:
get_filter("12345")

## End(Not run)
```

get_filter_id	<i>Get filter id by name</i>
---------------	------------------------------

Description

Get filter id by name

Usage

```
get_filter_id(
  filter_name,
  all_filters = get_all_filters(token = token),
  token = get_todoist_api_token()
)
```

Arguments

filter_name	name of the filter
all_filters	result of get_all_filters (optional)
token	todoist API token

Value

id of the filter

Examples

```
## Not run:  
get_filter_id("Urgent Today")  
  
## End(Not run)
```

get_label	<i>Get a single label by ID</i>
-----------	---------------------------------

Description

Get a single label by ID

Usage

```
get_label(label_id, token = get_todoist_api_token())
```

Arguments

label_id	id of the label
token	todoist API token

Value

list with label details

Examples

```
## Not run:  
get_label("12345")  
  
## End(Not run)
```

get_label_id *Get label id by name*

Description

Get label id by name

Usage

```
get_label_id(  
    label_name,  
    all_labels = get_all_labels(token = token),  
    token = get_todoist_api_token(),  
    create = TRUE  
)
```

Arguments

label_name	name of the label
all_labels	result of get_all_labels (optional)
token	todoist API token
create	boolean create label if needed

Value

id of the label

Examples

```
## Not run:  
get_label_id("urgent")  
  
## End(Not run)
```

get_productivity_stats
Get productivity stats

Description

Get productivity stats

Usage

```
get_productivity_stats(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

list with productivity statistics

Examples

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

get_project *Get a single project by ID*

Description

Get a single project by ID

Usage

```
get_project(project_id, token = get_todoist_api_token())
```

Arguments

project_id id of the project
token todoist API token

Value

list with project details

Examples

```
## Not run:  
get_project("12345")  
  
## End(Not run)
```

get_project_id	<i>Get id of project</i>
----------------	--------------------------

Description

This function gives you the id of a project by name, which is useful for adding tasks or people to the project.

Usage

```
get_project_id(  
  project_name,  
  all_projects = get_all_projects(token = token),  
  token = get_todoist_api_token(),  
  create = TRUE,  
  verbose = FALSE  
)
```

Arguments

project_name	name of the project
all_projects	result of get_all_projects
token	todoist API token
create	boolean create project if needed
verbose	boolean that make the function verbose

Value

id of project (character vector)

Examples

```
## Not run:  
get_all_projects() %>%  
  get_project_id("test")  
  
## End(Not run)
```

get_section	<i>Get a single section by ID</i>
-------------	-----------------------------------

Description

Get a single section by ID

Usage

```
get_section(section_id, token = get_todoist_api_token())
```

Arguments

section_id	id of the section
token	todoist API token

Value

list with section details

Examples

```
## Not run:  
get_section("12345")  
  
## End(Not run)
```

get_section_id	<i>Get section id</i>
----------------	-----------------------

Description

Get section id

Usage

```
get_section_id(  
  project_id = get_project_id(project_name = project_name, token = token),  
  project_name,  
  section_name,  
  token = get_todoist_api_token(),  
  all_section = get_section_from_project(project_id = project_id, token = token)  
)
```

Arguments

project_id	id of the project
project_name	name of the project
section_name	name of the section
token	token
all_section	all_section

Value

section id (character). Returns 0 if section not found.

get_shared_labels	<i>Get shared labels</i>
-------------------	--------------------------

Description

Get shared labels

Usage

```
get_shared_labels(token = get_todoist_api_token())
```

Arguments

token	todoist API token
-------	-------------------

Value

tibble of shared labels

Examples

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

get_task	<i>Get a single task by ID</i>
----------	--------------------------------

Description

Get a single task by ID

Usage

```
get_task(task_id, token = get_todoist_api_token())
```

Arguments

task_id	id of the task
token	todoist API token

Value

list with task details

Examples

```
## Not run:  
get_task("12345")  
  
## End(Not run)
```

get_tasks	<i>List of tasks</i>
-----------	----------------------

Description

List of tasks

Usage

```
get_tasks(token = get_todoist_api_token())
```

Arguments

token	todoist API token
-------	-------------------

Value

list of all tasks

Examples

```
## Not run:  
# Set API key first  
set_todoist_api_token()  
# Get all tasks  
tasks <- get_tasks()  
  
## End(Not run)
```

get_tasks_by_filter	<i>Get tasks by filter query</i>
---------------------	----------------------------------

Description

Get tasks by filter query

Usage

```
get_tasks_by_filter(query, token = get_todoist_api_token())
```

Arguments

query	filter query string (e.g., "today", "p1 & #Work")
token	todoist API token

Value

tibble of tasks matching the filter

Examples

```
## Not run:  
get_tasks_by_filter("today")  
get_tasks_by_filter("p1 & #Work")  
  
## End(Not run)
```

get_tasks_of_project *List of tasks of project*

Description

List of tasks of project

Usage

```
get_tasks_of_project(  
  project_id = get_project_id(project_name = project_name, token = token),  
  project_name,  
  token = get_todoist_api_token()  
)
```

Arguments

project_id	id of the project
project_name	name of the project
token	todoist API token

Value

list of all tasks

get_todoist_api_token *Get token stored by keyring*

Description

Return the todoist API token. If this is the first time, you will need to setup your token.

Usage

```
get_todoist_api_token(ask = TRUE)
```

Arguments

ask	booleen do we have to ask if missing
-----	--------------------------------------

Value

token (character vector)

Examples

```
## Not run:
get_todoist_api_token()

## End(Not run)
```

get_users_id	<i>Get users id</i>
--------------	---------------------

Description

Get users id

Usage

```
get_users_id(
  mails,
  all_users = get_all_users(token = token),
  token = get_todoist_api_token()
)
```

Arguments

mails	mails of the person
all_users	all_users
token	token

Value

id of users

get_users_in_project	<i>Get users in projects</i>
----------------------	------------------------------

Description

Get users in projects

Usage

```
get_users_in_project(
  project_id = get_project_id(project_name = project_name, token = token),
  project_name,
  token = get_todoist_api_token()
)
```

Arguments

project_id	id of the project
project_name	name of the project
token	token

Value

dataframe of users in projects

get_user_info	<i>Get current user info</i>
---------------	------------------------------

Description

Get current user info

Usage

```
get_user_info(token = get_todoist_api_token())
```

Arguments

token	todoist API token
-------	-------------------

Value

list with user information

Examples

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

get_workspace_users *Get workspace users*

Description

Get workspace users

Usage

```
get_workspace_users(token = get_todoist_api_token())
```

Arguments

token todoist API token

Value

tibble of workspace users

Examples

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

import_template *Import a template into a project*

Description

Import a template into a project

Usage

```
import_template(  
  project_id = get_project_id(project_name = project_name, token = token, create = FALSE),  
  project_name,  
  file_path,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

project_id	id of the project to import into
project_name	name of the project (for lookup if project_id not provided)
file_path	path to the template file (CSV format)
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:
import_template(project_name = "my_proj", file_path = "template.csv")

## End(Not run)
```

invite_to_workspace *Invite user to workspace*

Description

Invite user to workspace

Usage

```
invite_to_workspace(
  workspace_id,
  email,
  role = "member",
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

workspace_id	id of the workspace
email	email of the user to invite
role	role for the user (e.g., "member", "admin")
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
invite_to_workspace("12345", "user@example.com")  
  
## End(Not run)
```

leave_workspace	<i>Leave a workspace</i>
-----------------	--------------------------

Description

Leave a workspace

Usage

```
leave_workspace(workspace_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

workspace_id	id of the workspace
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
leave_workspace("12345")  
  
## End(Not run)
```

move_section	<i>Move a section to another project</i>
--------------	--

Description

Move a section to another project

Usage

```
move_section(  
    section_id,  
    project_id,  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

section_id	id of the section
project_id	target project id
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the moved section (invisible)

Examples

```
## Not run:  
move_section("12345", project_id = "67890")  
  
## End(Not run)
```

move_task

Move a task to another project or section

Description

Move a task to another project or section

Usage

```
move_task(  
    task_id,  
    project_id = NULL,  
    section_id = NULL,  
    parent_id = NULL,  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

task_id	id of the task to move
project_id	new project id (optional)
section_id	new section id (optional)
parent_id	new parent task id (optional)
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the moved task (invisible)

Examples

```
## Not run:  
move_task("12345", project_id = "67890")  
  
## End(Not run)
```

open_todoist_website_profile
Open todoist website

Description

Open todoist website

Usage

```
open_todoist_website_profile(verbose = TRUE)
```

Arguments

verbose	boolean that make the function verbose
---------	--

Value

open integration webpage from todoist website

Examples

```
open_todoist_website_profile()
```

quick_add_task	<i>Quick add a task using natural language</i>
----------------	--

Description

Quick add a task using natural language

Usage

```
quick_add_task(text, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

text	natural language text for the task
verbose	boolean that make the function verbose
token	todoist API token

Value

task object

Examples

```
## Not run:  
quick_add_task("Buy milk tomorrow at 5pm #Shopping")  
  
## End(Not run)
```

random_key	<i>Random key</i>
------------	-------------------

Description

Random key

Usage

```
random_key()
```

Value

key
random key generate with digest

reject_invitation *Reject a project invitation*

Description

Reject a project invitation

Usage

```
reject_invitation(  
    invitation_id,  
    invitation_secret,  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

invitation_id	id of the invitation
invitation_secret	secret of the invitation
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
reject_invitation("12345", "secret123")  
  
## End(Not run)
```

remove_shared_label *Remove a shared label*

Description

Remove a shared label

Usage

```
remove_shared_label(name, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

name	name of the shared label to remove
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:  
remove_shared_label("label_name")  
  
## End(Not run)
```

`rename_shared_label` *Rename a shared label*

Description

Rename a shared label

Usage

```
rename_shared_label(  
  old_name,  
  new_name,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

old_name	current name of the shared label
new_name	new name for the shared label
verbose	boolean that make the function verbose
token	todoist API token

Value

NULL (invisible)

Examples

```
## Not run:
rename_shared_label("old_name", "new_name")

## End(Not run)
```

reopen_task	<i>Reopen (uncomplete) a task</i>
-------------	-----------------------------------

Description

Reopen (uncomplete) a task

Usage

```
reopen_task(task_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

task_id	id of the task to reopen
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the reopened task (invisible)

Examples

```
## Not run:
reopen_task("12345")

## End(Not run)
```

set_todoist_api_token	<i>Set todoist API token</i>
-----------------------	------------------------------

Description

This function use keyring to store your token from your todoist profile. To find your token from todoist website, use [open_todoist_website_profile](#)

Usage

```
set_todoist_api_token(token)
```

Arguments

token todoist API token

Value

token

unarchive_project *Unarchive a project*

Description

Unarchive a project

Usage

```
unarchive_project(project_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

project_id id of the project
verbose boolean that make the function verbose
token todoist API token

Value

id of the unarchived project (invisible)

Examples

```
## Not run:  
unarchive_project(project_id = "12345")  
  
## End(Not run)
```

unarchive_section *Unarchive a section*

Description

Unarchive a section

Usage

```
unarchive_section(section_id, verbose = TRUE, token = get_todoist_api_token())
```

Arguments

section_id	id of the section
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the unarchived section (invisible)

Examples

```
## Not run:  
unarchive_section("12345")  
  
## End(Not run)
```

update_comment *Update a comment*

Description

Update a comment

Usage

```
update_comment(  
  comment_id,  
  content,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

comment_id	id of the comment
content	new content for the comment
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the updated comment (invisible)

Examples

```
## Not run:
update_comment("12345", content = "Updated comment")

## End(Not run)
```

update_filter	<i>Update a filter</i>
---------------	------------------------

Description

Update a filter

Usage

```
update_filter(
  filter_id = get_filter_id(filter_name = filter_name, token = token),
  filter_name,
  new_name = NULL,
  query = NULL,
  color = NULL,
  is_favorite = NULL,
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

filter_id	id of the filter
filter_name	name of the filter (for lookup if filter_id not provided)
new_name	new name for the filter
query	new query string
color	new color
is_favorite	boolean to mark as favorite
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the updated filter (invisible)

Examples

```
## Not run:
update_filter(filter_name = "Urgent Today", query = "today & p1 & #Work")

## End(Not run)
```

update_label	<i>Update a label</i>
--------------	-----------------------

Description

Update a label

Usage

```
update_label(
  label_id = get_label_id(label_name = label_name, token = token, create = FALSE),
  label_name,
  new_name = NULL,
  color = NULL,
  is_favorite = NULL,
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

label_id	id of the label
label_name	name of the label (for lookup if label_id not provided)
new_name	new name for the label
color	new color for the label
is_favorite	boolean to mark as favorite
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the updated label (invisible)

Examples

```
## Not run:
update_label(label_name = "urgent", new_name = "very_urgent")

## End(Not run)
```

update_project	<i>Update a project</i>
----------------	-------------------------

Description

Update a project

Usage

```
update_project(
  project_id = get_project_id(project_name = project_name, token = token, create = FALSE),
  project_name,
  new_name = NULL,
  color = NULL,
  is_favorite = NULL,
  view_style = NULL,
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

project_id	id of the project
project_name	name of the project (for lookup if project_id not provided)
new_name	new name for the project
color	new color for the project
is_favorite	boolean to mark as favorite
view_style	view style ("list" or "board")
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the updated project (invisible)

Examples

```
## Not run:
update_project(project_name = "my_proj", new_name = "new_name")

## End(Not run)
```

update_reminder	<i>Update a reminder</i>
-----------------	--------------------------

Description

Update a reminder

Usage

```
update_reminder(  
  reminder_id,  
  due_date = NULL,  
  due_datetime = NULL,  
  minute_offset = NULL,  
  type = NULL,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

reminder_id	id of the reminder
due_date	new due date for the reminder (format: YYYY-MM-DD)
due_datetime	new due datetime for the reminder (format: YYYY-MM-DDTHH:MM:SS)
minute_offset	new minutes before due time to remind
type	new type of reminder
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the updated reminder (invisible)

Examples

```
## Not run:  
update_reminder("12345", due_datetime = "2024-12-26T10:00:00")  
  
## End(Not run)
```

update_section	<i>Update a section</i>
----------------	-------------------------

Description

Update a section

Usage

```
update_section(  
    section_id,  
    new_name,  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

section_id	id of the section
new_name	new name for the section
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the updated section (invisible)

Examples

```
## Not run:  
update_section("12345", new_name = "New Section Name")  
  
## End(Not run)
```

update_task	<i>Update a task</i>
-------------	----------------------

Description

Update a task

Usage

```
update_task(  
    task_id,  
    content = NULL,  
    description = NULL,  
    priority = NULL,  
    due_string = NULL,  
    due_date = NULL,  
    labels = NULL,  
    verbose = TRUE,  
    token = get_todoist_api_token()  
)
```

Arguments

task_id	id of the task to update
content	new content/title for the task
description	new description for the task
priority	priority (1-4, 4 being highest)
due_string	due date as string (e.g., "tomorrow", "every monday")
due_date	due date as date (format: YYYY-MM-DD)
labels	vector of label names
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the updated task (invisible)

Examples

```
## Not run:  
update_task("12345", content = "Updated task name")  
  
## End(Not run)
```

update_todoist_api_token

Update Todoist Api Token

Description

Remove the old token and register a new one.

Usage

```
update_todoist_api_token()
```

Value

nothing, storing your token

update_workspace	<i>Update a workspace</i>
------------------	---------------------------

Description

Update a workspace

Usage

```
update_workspace(  
  workspace_id,  
  name,  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

workspace_id	id of the workspace
name	new name for the workspace
verbose	boolean that make the function verbose
token	todoist API token

Value

id of the updated workspace (invisible)

Examples

```
## Not run:  
update_workspace("12345", name = "New Workspace Name")  
  
## End(Not run)
```

`upload_file`*Upload a file*

Description

Upload a file

Usage

```
upload_file(  
  file_path,  
  file_name = basename(file_path),  
  verbose = TRUE,  
  token = get_todoist_api_token()  
)
```

Arguments

<code>file_path</code>	path to the file to upload
<code>file_name</code>	name for the uploaded file (defaults to basename of <code>file_path</code>)
<code>verbose</code>	boolean that make the function verbose
<code>token</code>	todoist API token

Value

upload object with `file_url`

Examples

```
## Not run:  
upload <- upload_file("document.pdf")  
# Use upload$file_url in a comment  
  
## End(Not run)
```

Index

accept_invitation, 4
add_comment, 4
add_filter, 5
add_label, 6
add_project, 7
add_reminder, 8
add_responsible_to_task, 9
add_section, 9
add_tasks_in_project, 10
add_tasks_in_project_from_df, 11
add_user_in_project, 13
add_users_in_project, 12
archive_project, 14
archive_section, 15
ask_todoist_api_token, 15

call_api, 16
call_api_rest, 16
close_task, 17

delete_collaborator, 17
delete_comment, 18
delete_filter, 19
delete_invitation, 19
delete_label, 20
delete_project, 21
delete_reminder, 22
delete_section, 22
delete_task, 23
delete_todoist_api_token, 24
delete_upload, 24
download_backup, 25

escape_json, 25
export_template, 26

get_activity_logs, 27
get_all_data, 28
get_all_filters, 28
get_all_labels, 29
get_all_projects, 30, 40
get_all_reminders, 30
get_all_sections, 31
get_all_users, 31
get_all_workspaces, 32
get_archived_projects, 33
get_backups, 33
get_comment, 34
get_comments, 34
get_completed_tasks, 35
get_filter, 36
get_filter_id, 36
get_label, 37
get_label_id, 38
get_productivity_stats, 38
get_project, 39
get_project_id, 40
get_section, 41
get_section_id, 41
get_shared_labels, 42
get_task, 43
get_tasks, 43
get_tasks_by_filter, 44
get_tasks_of_project, 45
get_todoist_api_token, 45
get_user_info, 47
get_users_id, 46
get_users_in_project, 46
get_workspace_users, 48

import_template, 48
invite_to_workspace, 49

leave_workspace, 50

move_section, 50
move_task, 51

open_todoist_website_profile, 52, 56
quick_add_task, 53

random_key, [53](#)
reject_invitation, [54](#)
remove_shared_label, [54](#)
rename_shared_label, [55](#)
reopen_task, [56](#)

set_todoist_api_token, [56](#)

unarchive_project, [57](#)
unarchive_section, [58](#)
update_comment, [58](#)
update_filter, [59](#)
update_label, [60](#)
update_project, [61](#)
update_reminder, [62](#)
update_section, [63](#)
update_task, [63](#)
update_todoist_api_token, [64](#)
update_workspace, [65](#)
upload_file, [66](#)