Package: rtodoist (via r-universe)

July 3, 2024

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

Version 0.1.9007

Description Allows you to interact with the API of the ``Todoist'' platform. 'Todoist' <https://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 digest, dplyr, getPass, glue, httr, httr2, keyring, magrittr, purrr, stats, stringi, stringr, utils

Suggests knitr, lubridate, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Repository https://thinkr-open.r-universe.dev

RemoteUrl https://github.com/ThinkR-open/rtodoist

RemoteRef HEAD

RemoteSha 291d22104c2bd2dc819dfd7a62dd90585a9cf659

Contents

`add_project` *Add a new project*

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_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

`project_id`	id of the project
`project_name`	name of the project
`responsible`	add someone to this task with mail
`task`	the full name of the task
`verbose`	boolean that make the function verbose
`all_users`	all_users
`token`	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

---

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)
```

`ask_todoist_api_token` *Pop-up to save the token*

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` *Call the good version of API*

Description

Call the good version of API

Usage

```
call_api(
  ...,
  url = "https://todoist.com/api/v9/sync",
  token = get_todoist_api_token()
)
```

Arguments

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

Value

list

---

call_api_project_data *Call project data*

Description

Call project data

Usage

```
call_api_project_data(
  ...,
  url = "https://api.todoist.com/sync/v9/projects/get_data"
)
```

Arguments

...	any params of POST request
url	url to call

Value

list

---

delete_todoist_api_token *Delete todoist api token*

Description

Delete todoist api token

Usage

```
delete_todoist_api_token()
```

Value

nothing, delete the api token

get_all_data *Get all objects inside a list*

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_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_users *Get users*

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_project_id *Get id of project*

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_id *get id section*

Description

get id section

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

---

get_tasks *List of tasks*

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_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` *Get users id*

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` *Get users in projects*

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

---

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()
```

---

random_key *Random key*

Description

Random key

Usage

```
random_key()
```

Value

key

random key generate with digest

set_todoist_api_token *Set todoist API token*

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

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

Index