Package ‘rtodoist’

October 14, 2022

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

Version 0.1.0

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

Depends R (>= 3.5.0)

Imports digest, dplyr, getPass, glue, httr, keyring, magrittr, purrr,
utils

Suggests knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

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

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

NeedsCompilation no

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

Maintainer Cervan Girard <cervan@thinkr.fr>

Repository CRAN

Date/Publication 2020-05-14 08:40:02 UTC

R topics documented:

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	make it talk
token	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,
  add_responsible,
  task,
  verbose,
  token = get_todoist_api_token()
)
```

Arguments

`project_id`	id of the project
`add_responsible`	add someone to this task with mail
`task`	the full name of the task
`verbose`	make the function verbose
`token`	token

Value

http request

`add_tasks_in_project` *Add a list of tasks*

Description

Add a list of tasks

Usage

```
add_tasks_in_project(
  project_id,
  tasks_list,
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token()
)
```

Arguments

`project_id`	id of project
`tasks_list`	list of tasks
`try_again`	start again the request
`time_try_again`	number of tries
`verbose`	make it talk
`responsible`	add people in project
`token`	token

Value

id of project (character vector)

See Also

[add_task_in_project()]

Examples

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

## End(Not run)
```

`add_task_in_project` *Add task in project*

Description

To work, it needs a project id.

Usage

```
add_task_in_project(
  project_id,
  task,
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token(),
  exiting_tasks = get_tasks(token = token)
)
```

Arguments

`project_id`	id of a project
`task`	a task
`try_again`	start again the request
`time_try_again`	number of tries
`verbose`	make it talk
`responsible`	add people in project with email. To know user email, use `get_users`.
`token`	token
`exiting_tasks`	list of tasks already in the project

Value

project id

Examples

```
## Not run:
add_project("my_proj") %>%
   add_task_in_project("Add tasks")

## End(Not run)
```

`add_users_in_project` *Add a list of users*

Description

Add a list of users

Usage

```
add_users_in_project(
  project_id,
  list_of_users,
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

`project_id`	id of project
`list_of_users`	list of mails
`verbose`	make it talk
`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,
  mail,
  verbose = TRUE,
  token = get_todoist_api_token()
)
```

Arguments

`project_id`	id of project
`mail`	mail of the user
`verbose`	make it talk
`token`	token

Value

id of project (character vector)

Examples

```
## Not run:
get_id_project("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(...)
```

Arguments

`...`	any params of POST request

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 *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(token = get_todoist_api_token())
```

Arguments

token	token

Value

list of all objects

Examples

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

## End(Not run)
```

get_id_project *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_id_project(object, project_name)
```

Arguments

`object`	result of get_projects
`project_name`	name of the project

Value

id of project (character vector)

Examples

```
## Not run:
get_projects() %>%
    get_id_project("test")

## End(Not run)
```

get_projects *List of projects*

Description

List of projects

Usage

```
get_projects(token = get_todoist_api_token())
```

Arguments

`token`	token

Value

list of all projects

Examples

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

## End(Not run)
```

---

get_tasks *List of tasks*

Description

List of tasks

Usage

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

Arguments

token	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_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 *Get users*

Description

Get a tibble with emails and ids of users

Usage

```
get_users(token = get_todoist_api_token())
```

Arguments

token	token

Value

tibble of users

Examples

```
## Not run:
get_users()

## End(Not run)
```

`get_user_id` *Get user id*

Description

Get user id

Usage

```
get_user_id(mail, token = get_todoist_api_token())
```

Arguments

`mail`	mail of the person
`token`	token

Value

id of user

`init_project` *Init project*

Description

Create a project with a specific to-do list for different people

Usage

```
init_project(
  project_id,
  tasks_list,
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token()
)
```

```
init_presta(
  project_id,
  tasks_list = list("Reunions", "Proposition - Devis", "Gestion projet", "Code",
    "S'assurer d'avoir un nom de projet coherent avec Slack",
    "S'assigner et mettre des dates sur certaines taches pour pas les oublier",
    "Facturer", "Rediger la reference de la mission dans {reference}", "Paye"),
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token()
)

init_presta_admin(
  project_id,
  tasks_list = list("Facturer", "Etre Pay\\u00E9"),
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token()
)

init_presta_manager(
  project_id,
  tasks_list = list("Proposition - Devis", "Gestion projet",
    "S'assurer d'avoir un nom de projet coherent avec Slack",
    "S'assigner et mettre des dates sur certaines taches pour pas les oublier",
    "Rediger la reference de la mission dans {reference}"),
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token()
)

init_forma_formateur(
  project_id,
  tasks_list = list("f_S'assurer d'avoir logistique OK",
   "f_Cr\\u00E9er le contenu de la formation", "f_Imprimer des feuilles d'emargement",
    "f_S'assurer d'avoir son materiel complet", "f_Remplir notes de frais",
    "f_Scanner feuille d'emargement", "f_transferer feuille d'emargement"),
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token()
)
```

```
init_forma_manager(
  project_id,
  tasks_list = list("Envoi du devis + contenu",
    "Assigner au formateur les taches du formateur", "Etre Pay\\u00E9"),
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token()
)

init_inter(
  project_id,
  tasks_list = list("S'assurer d'avoir un nom de projet coherent avec Slack",
    "S'assigner et mettre des dates sur certaines taches pour pas les oublier",
    "Envoyer les questionnaires", "Envoi du devis + contenu",
    "Envoi des conventions de formation", "Envoi des convocations",
    "Creer le contenu de la formation", "Reserver salle",
    "Envoi des liens d'installations", "Imprimer des feuilles d'emargement",
    "S'assurer d'avoir son materiel complet", "Reserver Resto",
    "Remplir ses notes de frais", "Scanner la feuille d'emargement",
    "Enregistrer son temps de formation dans Toggl",
    "Envoyer le questionnaire de satisfaction",
    "Envoyer questionnaire de satisfaction a froid", "Facturer", "Paye"),
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  token = get_todoist_api_token()
)

init_forma_admin(
  project_id,
  tasks_list = list("recevoir nom/prenom et mail des stagiaires",
    "envoyer test de pr\\u00E9\\u00E9valuation",
    "mettre les echeances aux taches suivantes", "envoyer convention",
    "v\\u00E9rifier retour convention",
    "s'assurer que la logistique du formateur est OK",
    "envoyer convocation aux stagiaires", "upload feuille emmargement",
   "envoyer facture", "verifier que la facture est bien dans le bon projet axonaut",
    "envoyer attestation", "envoyer evaluation a chaud",
    "plannifier l'evaluation a froid", "verifie retour eval a cahud",
    "verifie retour eval \\u00E0 froid",
    "mail retour de qualit\\u00E9 + audt besoin"),
  try_again = 3,
  time_try_again = 3,
  verbose = TRUE,
  responsible = NULL,
  token = get_todoist_api_token()
)
```

Arguments

project_id	id of project
tasks_list	lists of tasks
try_again	try again the http request
time_try_again	number of tries
verbose	make the function verbose
responsible	add people in project
token	token

Value

id of project (character vector)

open_todoist_website_profile

Open todoist website

Description

Open todoist website

Usage

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

Arguments

verbose	make it talk

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