

Package: clickrup (via r-universe)

August 31, 2024

Type Package

Title Interacting with the ClickUp v2 API from R

Version 0.0.6

Date 2021-12-12

Maintainer Peter Solymos <psolymos@gmail.com>

Description Work with the ClickUp productivity app from R to manage tasks, goals, time tracking, and more.

License MIT + file LICENSE

LazyLoad yes

LazyData true

Imports httr, jsonlite, rlang

Suggests knitr

URL <https://github.com/psolymos/clickrup>

BugReports <https://github.com/psolymos/clickrup/issues>

VignetteBuilder knitr

RoxygenNote 7.2.3

Encoding UTF-8

Roxygen list(markdown = TRUE)

Repository <https://psolymos.r-universe.dev>

RemoteUrl <https://github.com/psolymos/clickrup>

RemoteRef HEAD

RemoteSha 15f888195d8851258211102f2d232990944a271b

Contents

api-attachment	2
api-authorization	3
api-checklists	4

api-comments	6
api-customfields	7
api-dependencies	7
api-folders	8
api-goals	9
api-guests	10
api-lists	11
api-members	12
api-sharedhierarchy	13
api-spaces	14
api-tags	15
api-tasks	16
api-tasktemplates	18
api-teams	18
api-timetracking-2	19
api-timetracking-legacy	21
api-users	22
api-views	23
api-webhooks	24
cu-class	25
cu-date	26
cu-get-all-team-tasks	27
cu-options	27
cu-pat	28
cu-priority	29
cu-ratelimit	30
cu-response	31
cu-role	31
cu-task-id	32

Index 33

api-attachment	<i>Attachments</i>
----------------	--------------------

Description

Result from a call to the API endpoint POST https://api.clickup.com/api/v2/task/task_id/attachment

Usage

```
cu_post_task_attachment(
    task_id,
    attachment,
    filename = NULL,
    ...,
    cu_token = NULL
)
```

Arguments

task_id	Task ID.
attachment	Path to a file.
filename	File name as it will show up in CluckUp, deduced from attachment when not provided (NULL).
...	Can be used to pass mime type argument to <code>httr::upload_file()</code> , mime type is guessed otherwise.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.

Value

A cu object.

See Also

[cu](#) for the cu object class.

Examples

```
## Not run:
## produce a png image
f <- tempfile(fileext=".png")
png(f)
plot(1:10, col=1:10)
dev.off()

## attache the image to a task
cu_post_task_attachment("8ach57", f)

unlink(f) # clean up the image

## End(Not run)
```

api-authorization

Authorization

Description

These are the routes for authing the API and going through the OAuth flow. Applications utilizing the personal API token do not need these functions.

Usage

```
cu_get_access_token(client_id, client_secret, code, cu_token = NULL)
```

```
cu_get_authorized_user(cu_token = NULL)
```

```
cu_get_authorized_teams(cu_token = NULL)
```

Arguments

<code>client_id</code>	Oauth app client id.
<code>client_secret</code>	Oauth app client secret.
<code>code</code>	Code given in redirect URL.
<code>cu_token</code>	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-checklists

Task checklists

Description

Working with checklists in ClickUp tasks.

Usage

```
cu_create_checklist(task_id, name, cu_token = NULL)
```

```
cu_edit_checklist(checklist_id, position, cu_token = NULL)
```

```
cu_delete_checklist(checklist_id, cu_token = NULL)
```

```
cu_create_checklist_item(checklist_id, name, ..., cu_token = NULL)
```

```
cu_edit_checklist_item(checklist_id, checklist_item_id, ..., cu_token = NULL)
```

```
cu_delete_checklist_item(checklist_id, checklist_item_id, cu_token = NULL)
```

Arguments

task_id	Task ID.
name	Checklist or checklist item name.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
checklist_id	Checklist ID.
position	Position is the zero-based index of the order you want the checklist to exist on the task. If you want the checklist to be in the first position, pass 0.
...	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
checklist_item_id	Checklist item ID.

Value

A cu object.

See Also

[cu](#) for the cu object class.

Examples

```
## Not run:
## create new checklist
cl <- cu_create_checklist("8ach57", "New checklist")

## change position of checklist
cu_edit_checklist(cl$checklist$id, position = 0)

## create checklist items
cli <- cu_create_checklist_item(cl$checklist$id, name="New item 1")
cli <- cu_create_checklist_item(cl$checklist$id, name="New item 2",
  assignee = 4471793)
cli <- cu_create_checklist_item(cl$checklist$id, name="New item 3",
  assignee = NA, resolved = TRUE, parent = NA)

## delete checklist items
cu_delete_checklist_item(cli$checklist$id,
  cli$checklist$items[[1]]$id)

## delete checklist
cu_delete_checklist(cli$checklist$id)

## End(Not run)
```

api-comments

Comments

Description

Working with comments in ClickUp tasks.

Usage

```
cu_post_task_comment(task_id, ..., cu_token = NULL)
```

```
cu_post_view_comment(view_id, ..., cu_token = NULL)
```

```
cu_post_list_comment(list_id, ..., cu_token = NULL)
```

```
cu_get_task_comments(task_id, cu_token = NULL)
```

```
cu_get_view_comments(view_id, cu_token = NULL)
```

```
cu_get_list_comments(list_id, cu_token = NULL)
```

```
cu_put_update_comment(comment_id, ..., cu_token = NULL)
```

```
cu_delete_comment(comment_id, cu_token = NULL)
```

Arguments

<code>task_id</code>	Task ID.
<code>...</code>	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
<code>cu_token</code>	ClickUp personal access token or an access token from the OAuth flow. The <code>CU_PAT</code> environment variable is used when NULL.
<code>view_id</code>	View ID.
<code>list_id</code>	List ID.
<code>comment_id</code>	Comment ID.

Value

A `cu` object.

See Also

[cu](#) for the `cu` object class.

api-customfields	<i>Custom fields</i>
------------------	----------------------

Description

Working with custom fields in ClickUp tasks.

Usage

```
cu_get_accessible_custom_fields(list_id, cu_token = NULL)
```

```
cu_set_custom_field_value(task_id, field_id, ..., cu_token = NULL)
```

```
cu_remove_field_value(task_id, field_id, cu_token = NULL)
```

Arguments

<code>list_id</code>	List ID.
<code>cu_token</code>	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
<code>task_id</code>	Task ID.
<code>field_id</code>	Field ID.
<code>...</code>	Named arguments to be passed to API request body. The accessible fields can be found on the task object from the <code>cu_get_task()</code> route. This is where you can retrieve the <code>field_id</code> . see the ClickUp API documentation (https://clickup.com/api).

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-dependencies	<i>Task relationships</i>
------------------	---------------------------

Description

Working dependencies xyz in ClickUp tasks (<https://clickup.com/api>).

Usage

```
cu_add_dependency(task_id, depends_on, dependency_of, cu_token = NULL)
```

```
cu_delete_dependency(task_id, depends_on, dependency_of, cu_token = NULL)
```

```
cu_add_task_link(task_id, links_to, cu_token = NULL)
```

```
cu_delete_task_link(task_id, links_to, cu_token = NULL)
```

Arguments

<code>task_id</code>	Task ID.
<code>depends_on</code>	A waiting on dependency of the task. One and only one of <code>depends_on</code> or <code>dependency_of</code> must be passed.
<code>dependency_of</code>	A blocking dependency of the task. One and only one of <code>depends_on</code> or <code>dependency_of</code> must be passed.
<code>cu_token</code>	ClickUp personal access token or an access token from the OAuth flow. The <code>CU_PAT</code> environment variable is used when NULL.
<code>links_to</code>	Link to another task.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-folders

Folders

Description

Working with folders in ClickUp tasks (<https://clickup.com/api>).

Usage

```
cu_create_folder(space_id, name, cu_token = NULL)
```

```
cu_update_folder(folder_id, name, cu_token = NULL)
```

```
cu_delete_folder(folder_id, name, cu_token = NULL)
```

```
cu_get_folders(space_id, archived = FALSE, cu_token = NULL)
```

```
cu_get_folder(folder_id, cu_token = NULL)
```


Arguments

space_id	Space ID.
name	Name of the folder.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
folder_id	Folder ID.
archived	Logical, to return archived (TRUE) folders.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-goals

Goals

Description

Working with goals in ClickUp tasks.

Usage

```
cu_create_goal(team_id, ..., cu_token = NULL)
cu_update_goal(goal_id, ..., cu_token = NULL)
cu_delete_goal(goal_id, cu_token = NULL)
cu_get_goals(team_id, cu_token = NULL)
cu_get_goal(goal_id, cu_token = NULL)
cu_create_key_result(goal_id, ..., cu_token = NULL)
cu_edit_key_result(key_result_id, ..., cu_token = NULL)
cu_delete_key_result(key_result_id, cu_token = NULL)
```

Arguments

<code>team_id</code>	Team ID.
<code>...</code>	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
<code>cu_token</code>	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
<code>goal_id</code>	Goal ID.
<code>key_result_id</code>	Key result ID.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-guests

Guests

Description

Working with guests in ClickUp tasks (Enterprise only feature).

Usage

```
cu_invite_guest_to_workspace(team_id, cu_token = NULL)
```

```
cu_edit_guest_on_workspace(team_id, guest_id, ..., cu_token = NULL)
```

```
cu_remove_guest_from_workspace(team_id, guest_id, cu_token = NULL)
```

```
cu_get_guest(team_id, guest_id, cu_token = NULL)
```

```
cu_add_guest_to_task(
  task_id,
  guest_id,
  permission_level = "read",
  cu_token = NULL
)
```

```
cu_remove_guest_from_task(task_id, guest_id, cu_token = NULL)
```

```
cu_add_guest_to_list(
  list_id,
  guest_id,
```

```

    permission_level = "read",
    cu_token = NULL
)

cu_remove_guest_from_list(list_id, guest_id, cu_token = NULL)

cu_add_guest_to_folder(
  folder_id,
  guest_id,
  permission_level = "read",
  cu_token = NULL
)

cu_remove_guest_from_folder(folder_id, guest_id, cu_token = NULL)

```

Arguments

team_id	Team ID.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
guest_id	Guest ID.
...	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
task_id	Task ID.
permission_level	Permission level can be "read", "comment", "edit", or "create".
list_id	List ID.
folder_id	Folder ID.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-lists

Lists

Description

Working with lists in ClickUp tasks.

Usage

```

cu_create_list(folder_id, ..., cu_token = NULL)

cu_create_folderless_list(space_id, ..., cu_token = NULL)

cu_update_list(list_id, ..., cu_token = NULL)

cu_delete_list(list_id, ..., cu_token = NULL)

cu_get_lists(folder_id, archived = FALSE, cu_token = NULL)

cu_get_lists_folderless(space_id, archived = FALSE, cu_token = NULL)

cu_get_list(list_id, cu_token = NULL)

```

Arguments

folder_id	Folder ID.
...	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
space_id	SPace ID.
list_id	List ID.
archived	Logical, to return archived (TRUE) lists.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-members

Members

Description

Working with members in ClickUp tasks (<https://clickup.com/api>).

Usage

```

cu_get_task_members(task_id, cu_token = NULL)

cu_get_list_members(list_id, cu_token = NULL)

```

Arguments

task_id	Task ID.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
list_id	List ID.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-sharedhierarchy *Shared hierarchy*

Description

Working with shared hierarchy in ClickUp tasks (<https://clickup.com/api>). Returns all resources you have access to where you don't have access to its parent. For example, if you have access to a shared task, but don't have access to its parent list, it will come back in this request.

Usage

```
cu_get_shared(team_id, cu_token = NULL)
```

Arguments

team_id	Team ID.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-spaces

Spaces

Description

Working with spaces in ClickUp tasks.

Usage

```
cu_create_space(team_id, name, ..., cu_token = NULL)
```

```
cu_update_space(space_id, ..., cu_token = NULL)
```

```
cu_delete_space(space_id, cu_token = NULL)
```

```
cu_get_spaces(team_id, archived = FALSE, cu_token = NULL)
```

```
cu_get_space(space_id, cu_token = NULL)
```

Arguments

team_id	Team ID.
name	Name of the space.
...	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
space_id	Space ID.
archived	Logical, to return archived (TRUE) spaces.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-tags

Tags

Description

Working with tags in ClickUp tasks.

Usage

```
cu_get_space_tags(space_id, cu_token = NULL)
```

```
cu_create_space_tag(space_id, name, ..., cu_token = NULL)
```

```
cu_edit_space_tag(space_id, tag_name, cu_token = NULL)
```

```
cu_delete_space_tag(task_id, tag_name, cu_token = NULL)
```

```
cu_add_tag_to_task(task_id, tag_name, cu_token = NULL)
```

```
cu_delete_space_tag(task_id, tag_name, cu_token = NULL)
```

Arguments

space_id	Space ID.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
name	Tag name.
...	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
tag_name	Tag name.
task_id	Task ID.

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-tasks

*Tasks***Description**

Working with tasks in ClickUp tasks.

Usage

```
cu_create_task(list_id, ..., cu_token = NULL)
```

```
cu_update_task(task_id, ..., cu_token = NULL)
```

```
cu_delete_task(task_id, cu_token = NULL)
```

```
cu_get_tasks(list_id, archived = FALSE, ..., cu_token = NULL)
```

```
cu_get_task(task_id, cu_token = NULL)
```

```
cu_get_filtered_team_tasks(team_id, ..., cu_token = NULL)
```

Arguments

<code>list_id</code>	List ID.
<code>...</code>	Named arguments to be passed to API request body, of as query parameters, see the ClickUp API documentation (https://clickup.com/api).
<code>cu_token</code>	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
<code>task_id</code>	Task ID. The ClickUp GUI prepends task IDs with a leading hash, <code>cu_task_id()</code> is used internally to normalize task IDs.
<code>archived</code>	Logical, to return archived (TRUE) lists.
<code>team_id</code>	Team (workspace) ID.

Details**Create task:**

`assignees` is an array of the assignees' userids to be added to this task. `priority` is an integer mapping as 1 : Urgent, 2 : High, 3 : Normal, 4 : Low (`cu_priority()`). Time estimate is in milliseconds. See the fields section below to see the custom fields available on a list. If `notify_all` is true, creation notifications will be sent to everyone including the creator of the task. Instead of content you can pass `markdown_content` with valid markdown syntax to add formatting to the task description. Passing an existing task ID in the parent property will make the new task a subtask of that parent. The parent you pass must not be a subtask itself, and must be part of the list that you pass in the URL params. The `links_to` accepts a task ID to create a linked dependency on the new task.

Update task:

Only pass properties you want to update. It is unnecessary to pass the entire task object.

Get tasks:

The maximum number of tasks returned in this response is 100. When you are paging this request, you should check list limit against the length of each response to determine if you are on the last page. By default this does not include closed tasks. To page tasks, pass the page number you wish to fetch. `cu_get_all_team_tasks()` is a wrapper to spare paging.

Filter query parameters:

When value is an array, use `I()` to make sure atomic vectors remain JSON arrays in the output.

- archived: return archived tasks.
- page: page to fetch (starts at 0)
- order_by: order by field, defaults to created; options: id, created, updated, due_date.
- reverse: reverse order.
- subtasks: include subtasks, default false.
- space_ids: spaces to query, array.
- project_ids: projects to query, array.
- statuses: statuses to query, array.
- include_closed: by default, the api does not include closed tasks. Set this to true and don't send a status filter to include closed tasks.
- assignees: assignees to query, array.
- due_date_gt: filter due date greater than posix time.
- due_date_lt: filter due date less than posix time.
- date_created_gt: filter date created greater than posix time.
- date_created_lt: filter date created less than posix time.
- date_updated_gt: filter date updated greater than posix time.
- date_updated_lt: filter date updated less than posix time
- custom_fields: filter by custom fields. Must be a stringified JSON array of objects. See the custom fields section of the API docs for more details.

Value

A cu object.

See Also

`cu` for the cu object class, `cu_date_to()` and `cu_date_from()` to work with POSIX dates, `cu_get_all_team_tasks()` is a wrapper. `cu_task_id()` to remove leading hashes from task IDs.

api-tasktemplates *Task templates*

Description

Working with task templates in ClickUp tasks.

Usage

```
cu_get_task_templates(team_id, page, cu_token = NULL)
```

```
cu_create_task_from_template(list_id, template_id, name, ..., cu_token = NULL)
```

Arguments

team_id	Team ID.
page	Page to fetch (starts at 0). To page task templates, pass the page number you wish to fetch.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
list_id	List ID.
template_id	Template ID.
name	Name of the task.
...	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-teams *Teams (workspaces)*

Description

Working with teams (workspaces) in ClickUp tasks (<https://clickup.com/api>). Teams is the legacy term for what are now called workspaces in ClickUp. For compatablity, the term team is still used in the API v2. This is NOT the new 'Teams' feature which represents a group of users.

Usage

```
cu_get_teams(cu_token = NULL)
```

Arguments

`cu_token` ClickUp personal access token or an access token from the OAuth flow. The `CU_PAT` environment variable is used when NULL.

Value

A cu object.

See Also

[cu](#) for the cu object class.

Examples

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

api-timetracking-2 *Time tracking 2.0*

Description

Working with time tracking in ClickUp tasks. Note: A time entry that has a negative duration means that timer is currently ongoing for that user.

Usage

```
cu_get_time_entries_within_date_range(  
  team_id,  
  start_date,  
  end_date,  
  assignee,  
  cu_token = NULL  
)  
  
cu_get_singular_time_entry(team_id, timer_id, cu_token = NULL)  
  
cu_get_time_entry_history(team_id, timer_id, cu_token = NULL)  
  
cu_get_running_time_entry(team_id, timer_id, cu_token = NULL)
```

```
cu_create_time_entry(team_id, ..., cu_token = NULL)
cu_remove_tags_from_time_entries(team_id, ..., cu_token = NULL)
cu_get_all_tags_from_time_entries(team_id, cu_token = NULL)
cu_add_tags_from_time_entries(team_id, ..., cu_token = NULL)
cu_change_tag_names_from_time_entries(team_id, ..., cu_token = NULL)
cu_start_time_entry(team_id, timer_id, ..., cu_token = NULL)
cu_stop_time_entry(team_id, cu_token = NULL)
cu_delete_time_entry(team_id, timer_id, cu_token = NULL)
cu_update_time_entry(team_id, timer_id, ..., cu_token = NULL)
```

Arguments

<code>team_id</code>	Team ID.
<code>start_date</code>	POSIX start time in milliseconds.
<code>end_date</code>	POSIX end time in milliseconds.
<code>assignee</code>	User ids to filter by separated by commas. Note: Only Workspace Owners/Admins have access to do this.
<code>cu_token</code>	ClickUp personal access token or an access token from the OAuth flow. The <code>CU_PAT</code> environment variable is used when NULL.
<code>timer_id</code>	Timer ID.
<code>...</code>	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).

Value

A cu object.

See Also

[api-timetracking-legacy](#) for Time tracking legacy API, [cu](#) for the cu object class, [cu_date_from\(\)](#) and [cu_date_to\(\)](#) to deal with POSIX times.

api-timetracking-legacy

Time tracking legacy

Description

Working with time tracking in ClickUp tasks. Note: This is the legacy API for time tracking. Please use the new Time tracking API. You won't be able to update timer entries that uses our Time Tracking 2.0 features that uses labels, descriptions, and making time entries billable.

Usage

```
cu_track_time(task_id, ..., cu_token = NULL)
```

```
cu_get_tracked_time(task_id, cu_token = NULL)
```

```
cu_edit_time_tracked(task_id, interval_id, ..., cu_token = NULL)
```

```
cu_delete_time_tracked(task_id, interval_id, cu_token = NULL)
```

Arguments

task_id	Task ID.
...	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api). Edit the start, end, or total time of a time tracked entry.
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
interval_id	Interval ID.

Value

A cu object. Note: the legacy API comes with a deprecation warning.

See Also

[api-timetracking-2](#) for Time Tracking 2.0, [cu](#) for the cu object class, [cu_date_from\(\)](#) and [cu_date_to\(\)](#) to deal with POSIX times.

api-users

*Users***Description**

Working with users in ClickUp tasks (Enterprise only feture).

Usage

```
cu_invite_user_to_workspace(
    team_id,
    email,
    admin = FALSE,
    ...,
    cu_token = NULL
)
```

```
cu_edit_user_on_workspace(
    team_id,
    user_id,
    username,
    admin = FALSE,
    ...,
    cu_token = NULL
)
```

```
cu_remove_user_from_workspace(team_id, user_id, cu_token = NULL)
```

```
cu_get_user(team_id, user_id, cu_token = NULL)
```

Arguments

<code>team_id</code>	Team (workspace) ID.
<code>email</code>	Email.
<code>admin</code>	Logical, admin account or not
<code>...</code>	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
<code>cu_token</code>	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
<code>user_id</code>	User ID.
<code>username</code>	User name.

Value

A cu object.

See Also

[cu](#) for the cu object class.

 api-views

 Views

Description

Working views in ClickUp tasks.

Usage

```

cu_create_team_view(team_id, name, ..., cu_token = NULL)

cu_create_space_view(space_id, name, ..., cu_token = NULL)

cu_create_folder_view(folder_id, name, ..., cu_token = NULL)

cu_create_list_view(list_id, name, ..., cu_token = NULL)

cu_get_team_views(team_id, cu_token = NULL)

cu_get_space_views(space_id, cu_token = NULL)

cu_get_folder_views(folder_id, cu_token = NULL)

cu_get_list_views(list_id, cu_token = NULL)

cu_get_view(view_id, cu_token = NULL)

cu_get_view_tasks(view_id, page, cu_token = NULL)

cu_update_view(view_id, ..., cu_token = NULL)

cu_delete_view(view_id, cu_token = NULL)

```

Arguments

team_id	Team (workspace) ID.
name	View name.
...	Named arguments to be passed to API request body, see the ClickUp API documentation (https://clickup.com/api).
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
space_id	Space ID.

folder_id	Folder ID.
list_id	List ID.
view_id	View ID.
page	Page to fetch (starts at 0).

Value

A cu object.

See Also

[cu](#) for the cu object class.

api-webhooks	<i>Webhooks</i>
--------------	-----------------

Description

Working with webhooks in ClickUp tasks.

Usage

```
cu_create_webhook(team_id, endpoint, events = "*", ..., cu_token = NULL)
```

```
cu_update_webhook(webhook_id, ..., cu_token = NULL)
```

```
cu_delete_webhook(webhook_id, cu_token = NULL)
```

```
cu_get_webhooks(team_id, cu_token = NULL)
```

Arguments

team_id	Team ID.
endpoint	Webhook endpoint.
events	Events (see Details).
...	Named arguments to be passed to API request body, e.g. "endpoint", "events", or "status", see the ClickUp API documentation (https://clickup.com/api).
cu_token	ClickUp personal access token or an access token from the OAuth flow. The CU_PAT environment variable is used when NULL.
webhook_id	Webhook ID.

Details

You may filter the location of resources that get sent to a webhook by passing an optional `space_id`, `folder_id`, `list_id`, or `task_id` in the body of the request. Without specifying any events, all event types will be sent to the webhook. However, you can filter for specific actions by sending an events array. To subscribe to specific events, pass an array of events that you want to subscribe to, otherwise pass "*" to subscribe to everything.

Events include: "taskCreated" "taskUpdated" "taskDeleted" "listCreated" "listUpdated" "listDeleted" "folderCreated" "folderUpdated" "folderDeleted" "spaceCreated" "spaceUpdated" "spaceDeleted" "goalCreated" "goalUpdated" "goalDeleted" "keyResultCreated" "keyResultUpdated" "keyResultDeleted".

Value

A cu object.

See Also

[cu](#) for the cu object class. [cu_task_id\(\)](#) to remove leading hashes from task IDs.

cu-class

Methods for cu objects.

Description

Calls to the ClickUp API return objects of class cu. The object is a list based on the parsed JSON response from the call. The response is stored as an attribute. Methods are defined for better printing and coercion.

Usage

```
## S3 method for class 'cu'
as.list(x, ...)
```

```
## S3 method for class 'cu'
print(x, ...)
```

```
## S3 method for class 'cu'
str(object, ...)
```

Arguments

x	An object of class cu.
...	Other optional arguments passed to methods.
object	An object of class cu.

See Also

[cu_response\(\)](#) for retrieving the response attribute from a cu object.

cu-date

Handling ClickUp date/time values.

Description

cu_date_from turns ClickUp date/time format (Unix time in milliseconds) to `POSIXct()`. cu_date_to turns a `POSIXct` coercible date/time value back to Unix time.

Usage

```
cu_date_from(ms, tz = NULL, ...)
```

```
cu_date_to(dt)
```

Arguments

ms	ClickUp date/time format, Unix time in milliseconds.
tz	time zone, when NULL it defaults to <code>cu_options()</code> \$tz.
...	Other optional arguments passed to <code>as.POSIXct()</code> .
dt	<code>POSIXct</code> coercible date/time value.

Value

cu_date_from returns `POSIXct()`.
cu_date_to returns Unix time as character .

See Also

`cu_options()` to set tz globally.

Examples

```
## Not run:
task_id <- "8ach57"
due_date <- as.POSIXct("2020-12-24 21:15:49 MDT")

task <- cu_get_task(task_id)
task$due_date

cu_date_to(due_date)
task_up <- cu_update_task("8ach57", due_date=cu_date_to(due_date))

cu_date_from(task_up$due_date)
identical(cu_date_from(task_up$due_date), due_date)

## End(Not run)
```

cu-get-all-team-tasks *Wrapper function to get all tasks from a workspace*

Description

The wrapper function uses [cu_get_filtered_team_tasks\(\)](#) but takes care of paging.

Usage

```
cu_get_all_team_tasks(team_id, ...)
```

Arguments

<code>team_id</code>	Team ID of tasks.
<code>...</code>	All query parameters for cu_get_filtered_team_tasks() except for page (paging is taken care of), e.g. <code>include_closed</code> to include closed tasks.

Value

A list parsed from the JSON response. The return object does not have a response attribute.

See Also

[cu](#) for the cu object class.

cu-options *Global options for the clickrup package*

Description

Options store and allow to set global values for the ClickUp API and for clickrup functions.

Usage

```
cu_options(...)
```

Arguments

<code>...</code>	Options to set.
------------------	-----------------

Details

Options are mostly for internal use and to allow API version changes in the future.

The `tz` (time zone) option can be set globally for [cu_date_from\(\)](#). The default value `""` means current time zone, see [strptime\(\)](#).

Value

When parameters are set by `cu_options`, their former values are returned in an invisible named list. Such a list can be passed as an argument to `cu_options` to restore the parameter values. Tags are the following:

- `baseurl`: ClickUp API base URL.
- `version`: ClickUp API version.
- `tz`: time zone.
- `useragent`: user agent.

Examples

```
str(cu_options())

op <- cu_options(tz = "GMT") # save old values and set tz
cu_options()$tz             # new tz value
cu_options(op)              # reset
cu_options()$tz             # default tz value
```

 cu-pat

Managing ClickUp personal access token

Description

The ClickUp API requires a personal access token (PAT) to be sent with each API request. The `cu_set_pat` function uses `Sys.setenv` to set the `CU_PAT` environment variable for other processes called from within R or future calls to `Sys.getenv` from the same R process. `cu_get_pat` is used to retrieve the PAT for the API calls.

Usage

```
cu_get_pat(token = NULL)

cu_set_pat(token)
```

Arguments

`token` ClickUp personal access token or an access token from the OAuth flow. The `CU_PAT` environment variable is used when `NULL`.

Details

To set up a ClickUp PAT, Follow this [tutorial](#):

- sign up for ClickUp (you can use this referral [link](#) to do so, it's free),
- navigate to your personal *Settings*,
- click *Apps* in the left sidebar,
- click *Generate* to create your API token,
- click *Copy* to copy the token to your clipboard.

Now add your ClickUp token as an environment variable:

- open the file `.Renviro`: `file.edit("~/Renviro")`,
- add a line with `CU_PAT="your_token"` to the `.Renviro` file and save it,
- check with `Sys.getenv("CU_PAT")`, it should return the token.

Value

`cu_set_pat` returns logical similarly to [Sys.setenv\(\)](#).

`cu_get_pat` returns the ClickUp PAT that will look something like `pk_4753994_EXP7MPOJ7XQM5UJDV2M45MPF0YHH5YHO`. When token is not NULL it will simply pass through the token value and not look for the `CU_PAT` environment variable.

See Also

[Sys.setenv\(\)](#) and [Sys.getenv\(\)](#)

cu-priority

Turn ClickUp API priority scores to labels

Description

The ClickUp API returns priority values as integer scores. The `cu_priority` function maps these scores to the priority labels Urgent (1), High (2), Normal (3), Low (4).

Usage

```
cu_priority(score)
```

Arguments

`score` Character or numeric values between 1 and 4.

Value

Returns a character vector with the labels.

Examples

```
cu_priority(c(1, 2, 3, 4))
```

cu-ratelimit	<i>ClickUp API rate limit</i>
--------------	-------------------------------

Description

The ClickUp API is rate limited per OAuth/PAT token. You will receive a 429 HTTP status code if you exceed the rate limit. According to the API description (<https://clickup.com/api>), the rate limit is 100 requests per minute per token and is subject to change. At the time of writing, the limit is 900/min/token.

Usage

```
cu_ratelimit(x)
```

Arguments

x A cu object returned by an API call.

Value

Returns a list with the following elements:

- limit: rate limit (per minute),
- remaining: remaining requests.

See Also

[cu](#) for the cu object class.

Examples

```
## Not run:  
task <- cu_get_task("8ach57")  
cu_ratelimit(task)  
  
## End(Not run)
```

cu-response	<i>Extracts the response attribute</i>
-------------	----------------------------------------

Description

The function extracts the response attribute from a cu object after an ClickUp API call.

Usage

```
cu_response(x)
```

Arguments

x An object of class cu.

Value

A response object as returned by [httr::response\(\)](#). Only calls to existing API endpoints return a response attribute. Wrapper functions that make multiple calls to endpoints (e.g. due to paging etc.) do not store the responses and return NULL.

See Also

[httr::response\(\)](#) for the response object, and [cu](#) for the cu object class.

Examples

```
## Not run:
task <- cu_get_task("8ach57")
cu_response(task)

## End(Not run)
```

cu_role	<i>Turn ClickUp API role to meaningful label</i>
---------	--------------------------------------------------

Description

The ClickUp API returns role as integer. The `cu_role` function maps these scores to the priority labels Owner (1), Admin (2), Member (3), Guest (4).

Usage

```
cu_role(role)
```

Arguments

role Character or numeric values between 1 and 4.

Value

Returns a character vector with the labels.

Examples

```
cu_role(c(1, 2, 3, 4))
```

cu-task-id *Handling task IDs*

Description

Task IDs are prepended with a hash when copied from the ClickUp GUI. The function checks and removes the leading hash.

Usage

```
cu_task_id(task_id)
```

Arguments

task_id Task ID with or without a leading hash.

Value

Returns task ID as character without leading hash to be used in the API.

Examples

```
cu_task_id("#7ygh8h")  
cu_task_id("7ygh8h")
```


Index

api-attachment, 2
api-authorization, 3
api-checklists, 4
api-comments, 6
api-customfields, 7
api-dependencies, 7
api-folders, 8
api-goals, 9
api-guests, 10
api-lists, 11
api-members, 12
api-sharedhierarchy, 13
api-spaces, 14
api-tags, 15
api-tasks, 16
api-tasktemplates, 18
api-teams, 18
api-timetracking-2, 19, 21
api-timetracking-legacy, 20, 21
api-users, 22
api-views, 23
api-webhooks, 24
as.list.cu (cu-class), 25
as.POSIXct(), 26

cu, 3–15, 17–21, 23–25, 27, 30, 31
cu-class, 25
cu-date, 26
cu-get-all-team-tasks, 27
cu-options, 27
cu-pat, 28
cu-priority, 29
cu-ratelimit, 30
cu-response, 31
cu-role, 31
cu-task-id, 32
cu_add_dependency (api-dependencies), 7
cu_add_guest_to_folder (api-guests), 10
cu_add_guest_to_list (api-guests), 10
cu_add_guest_to_task (api-guests), 10
cu_add_tag_to_task (api-tags), 15
cu_add_tags_from_time_entries
 (api-timetracking-2), 19
cu_add_task_link (api-dependencies), 7
cu_change_tag_names_from_time_entries
 (api-timetracking-2), 19
cu_create_checklist (api-checklists), 4
cu_create_checklist_item
 (api-checklists), 4
cu_create_folder (api-folders), 8
cu_create_folder_view (api-views), 23
cu_create_folderless_list (api-lists),
 11
cu_create_goal (api-goals), 9
cu_create_key_result (api-goals), 9
cu_create_list (api-lists), 11
cu_create_list_view (api-views), 23
cu_create_space (api-spaces), 14
cu_create_space_tag (api-tags), 15
cu_create_space_view (api-views), 23
cu_create_task (api-tasks), 16
cu_create_task_from_template
 (api-tasktemplates), 18
cu_create_team_view (api-views), 23
cu_create_time_entry
 (api-timetracking-2), 19
cu_create_webhook (api-webhooks), 24
cu_date_from (cu-date), 26
cu_date_from(), 17, 20, 21, 27
cu_date_to (cu-date), 26
cu_date_to(), 17, 20, 21
cu_delete_checklist (api-checklists), 4
cu_delete_checklist_item
 (api-checklists), 4
cu_delete_comment (api-comments), 6
cu_delete_dependency
 (api-dependencies), 7
cu_delete_folder (api-folders), 8
cu_delete_goal (api-goals), 9

- cu_delete_key_result (api-goals), 9
- cu_delete_list (api-lists), 11
- cu_delete_space (api-spaces), 14
- cu_delete_space_tag (api-tags), 15
- cu_delete_task (api-tasks), 16
- cu_delete_task_link (api-dependencies), 7
- cu_delete_time_entry (api-timetracking-2), 19
- cu_delete_time_tracked (api-timetracking-legacy), 21
- cu_delete_view (api-views), 23
- cu_delete_webhook (api-webhooks), 24
- cu_edit_checklist (api-checklists), 4
- cu_edit_checklist_item (api-checklists), 4
- cu_edit_guest_on_workspace (api-guests), 10
- cu_edit_key_result (api-goals), 9
- cu_edit_space_tag (api-tags), 15
- cu_edit_time_tracked (api-timetracking-legacy), 21
- cu_edit_user_on_workspace (api-users), 22
- cu_get_access_token (api-authorization), 3
- cu_get_accessible_custom_fields (api-customfields), 7
- cu_get_all_tags_from_time_entries (api-timetracking-2), 19
- cu_get_all_team_tasks (cu-get-all-team-tasks), 27
- cu_get_all_team_tasks(), 17
- cu_get_authorized_teams (api-authorization), 3
- cu_get_authorized_user (api-authorization), 3
- cu_get_filtered_team_tasks (api-tasks), 16
- cu_get_filtered_team_tasks(), 27
- cu_get_folder (api-folders), 8
- cu_get_folder_views (api-views), 23
- cu_get_folders (api-folders), 8
- cu_get_goal (api-goals), 9
- cu_get_goals (api-goals), 9
- cu_get_guest (api-guests), 10
- cu_get_list (api-lists), 11
- cu_get_list_comments (api-comments), 6
- cu_get_list_members (api-members), 12
- cu_get_list_views (api-views), 23
- cu_get_lists (api-lists), 11
- cu_get_lists_folderless (api-lists), 11
- cu_get_pat (cu-pat), 28
- cu_get_running_time_entry (api-timetracking-2), 19
- cu_get_shared (api-sharedhierarchy), 13
- cu_get_singular_time_entry (api-timetracking-2), 19
- cu_get_space (api-spaces), 14
- cu_get_space_tags (api-tags), 15
- cu_get_space_views (api-views), 23
- cu_get_spaces (api-spaces), 14
- cu_get_task (api-tasks), 16
- cu_get_task(), 7
- cu_get_task_comments (api-comments), 6
- cu_get_task_members (api-members), 12
- cu_get_task_templates (api-tasktemplates), 18
- cu_get_tasks (api-tasks), 16
- cu_get_team_views (api-views), 23
- cu_get_teams (api-teams), 18
- cu_get_time_entries_within_date_range (api-timetracking-2), 19
- cu_get_time_entry_history (api-timetracking-2), 19
- cu_get_tracked_time (api-timetracking-legacy), 21
- cu_get_user (api-users), 22
- cu_get_view (api-views), 23
- cu_get_view_comments (api-comments), 6
- cu_get_view_tasks (api-views), 23
- cu_get_webhooks (api-webhooks), 24
- cu_invite_guest_to_workspace (api-guests), 10
- cu_invite_user_to_workspace (api-users), 22
- cu_options (cu-options), 27
- cu_options(), 26
- cu_post_list_comment (api-comments), 6
- cu_post_task_attachment (api-attachment), 2
- cu_post_task_comment (api-comments), 6
- cu_post_view_comment (api-comments), 6
- cu_priority (cu-priority), 29
- cu_priority(), 16
- cu_put_update_comment (api-comments), 6

`cu_ratelimit` (`cu-ratelimit`), 30
`cu_remove_field_value`
 (`api-customfields`), 7
`cu_remove_guest_from_folder`
 (`api-guests`), 10
`cu_remove_guest_from_list` (`api-guests`),
 10
`cu_remove_guest_from_task` (`api-guests`),
 10
`cu_remove_guest_from_workspace`
 (`api-guests`), 10
`cu_remove_tags_from_time_entries`
 (`api-timetracking-2`), 19
`cu_remove_user_from_workspace`
 (`api-users`), 22
`cu_response` (`cu-response`), 31
`cu_response()`, 25
`cu_role` (`cu-role`), 31
`cu_set_custom_field_value`
 (`api-customfields`), 7
`cu_set_pat` (`cu-pat`), 28
`cu_start_time_entry`
 (`api-timetracking-2`), 19
`cu_stop_time_entry`
 (`api-timetracking-2`), 19
`cu_task_id` (`cu-task-id`), 32
`cu_task_id()`, 16, 17, 25
`cu_track_time`
 (`api-timetracking-legacy`), 21
`cu_update_folder` (`api-folders`), 8
`cu_update_goal` (`api-goals`), 9
`cu_update_list` (`api-lists`), 11
`cu_update_space` (`api-spaces`), 14
`cu_update_task` (`api-tasks`), 16
`cu_update_time_entry`
 (`api-timetracking-2`), 19
`cu_update_view` (`api-views`), 23
`cu_update_webhook` (`api-webhooks`), 24

`httr::response()`, 31
`httr::upload_file()`, 3

`POSIXct()`, 26
`print.cu` (`cu-class`), 25

`str.cu` (`cu-class`), 25
`strptime()`, 27
`Sys.getenv()`, 29
`Sys.setenv()`, 29