

Package ‘qiitr’

May 9, 2026

Type Package

Title R Interface to Qiita API

Version 0.1.1

Description Qiita is a technical knowledge sharing and collaboration platform for programmers.
See <<https://qiita.com/api/v2/docs>> for more information.

URL <https://github.com/yutannihilation/qiitr>

BugReports <https://github.com/yutannihilation/qiitr/issues>

Depends R (>= 2.10)

Imports htr, jsonlite, purrr, rstudioapi

LazyData TRUE

Encoding UTF-8

Suggests testthat

RoxygenNote 7.1.0

License MIT + file LICENSE

NeedsCompilation no

Author Hiroaki Yutani [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-3385-7233>>)

Maintainer Hiroaki Yutani <yutani.ini@gmail.com>

Repository CRAN

Date/Publication 2020-05-16 05:40:02 UTC

Contents

qiita_api	2
qiita_auth	2
qiita_comment	3
qiita_item	4
qiita_tag	6
qiita_user	7
qiita_util	8

Index**10**

qiita_api	<i>Send A Request To Qiita API</i>
-----------	------------------------------------

Description

An [httr](#) Wrapper for Qiita API.

Usage

```
qiita_api(
  verb,
  path,
  payload = NULL,
  query = NULL,
  per_page = 100L,
  page_offset = 0L,
  page_limit = 1L,
  .expect204 = FALSE
)
```

Arguments

verb	Method type (e.g. GET, POST, DELETE).
path	Path of API.
payload	JSON payload.
query	Query strings.
per_page	Number of entries (e.g. items, tags, users) in one page.
page_offset	Number of offset pages.
page_limit	Max number of pages to retrieve.
.expect204	If TRUE, return TRUE for status code 204 and FALSE for status code 404.

qiita_auth	<i>Authorization with Qiita</i>
------------	---------------------------------

Description

qiitr uses QIITA_ACCESSTOKEN environmental variable for authorization. You can issue a personal access token at <https://qiita.com/settings/applications>. To set the envvar permanently, write QIITA_ACCESSTOKEN = (your access token) to a file and save it as .Renviron on current directory or the home directory (For more info, see [Startup](#)). To set the variable temporarily, use qiita_set_accesstoken.

Usage

```
qiita_set_accesstoken()
```

`qiita_comment`*Qiita Comments API*

Description

Get, write, update or delete comments via Qiita API.

Usage

```
qiita_get_comments(  
  comment_id = NULL,  
  item_id = NULL,  
  per_page = 100L,  
  page_offset = 0L,  
  page_limit = 1L  
)  
  
qiita_delete_comment(comment_id)  
  
qiita_update_comment(comment_id, body)  
  
qiita_post_comment(item_id, body)
```

Arguments

<code>comment_id</code>	Comment ID.
<code>item_id</code>	Item (article) ID.
<code>per_page</code>	Number of items per one page.
<code>page_offset</code>	Number of offset pages.
<code>page_limit</code>	Max number of pages to retrieve.
<code>body</code>	body of the item

Examples

```
## Not run:  
# get a comment by id  
qiita_get_comments(comment_id = "1fdbb164e19d79e10203")  
  
# get comments by item id  
qiita_get_comments(item_id = "b4130186e1e095719dcb")  
  
# post a comment to some item  
qiita_post_comment(item_id = "123456789", body = "Thank you!!!")  
  
## End(Not run)
```

`qiita_item`*Qiita Item (Article) API*

Description

Get, post, delete, stock or unstock articles via Qiita API.

Usage

```
qiita_get_items(  
  item_id = NULL,  
  tag_id = NULL,  
  user_id = NULL,  
  query = NULL,  
  per_page = 100L,  
  page_offset = 0L,  
  page_limit = 1L  
)
```

```
qiita_post_item(  
  title,  
  body,  
  tags = qiita_util_tag("R"),  
  coediting = FALSE,  
  private = TRUE,  
  gist = FALSE,  
  tweet = FALSE  
)
```

```
qiita_delete_item(item_id)
```

```
qiita_update_item(  
  item_id,  
  title,  
  body,  
  tags = list(qiita_util_tag("R")),  
  private = TRUE  
)
```

```
qiita_stock_item(item_id)
```

```
qiita_unstock_item(item_id)
```

```
qiita_is_stocked_item(item_id)
```

```
qiita_get_stocks(user_id, per_page = 100L, page_offset = 0L, page_limit = 1L)
```

Arguments

item_id	Item (Article) ID.
tag_id	Tag IDs (e.g. "R").
user_id	User ID (e.g. "yutannihilation").
query	Query string (e.g. "dplyr user:yutannihilation").
per_page	Number of items per one page.
page_offset	Number of offset pages.
page_limit	Max number of pages to retrieve.
title	Title.
body	Content body.
tags	Tags. Use qiita_util_tag to generate tag objects.
coediting	If TRUE, the post will be editable by team members.
private	If TRUE, the post will be private.
gist	If TRUE, post the code to Gist.
tweet	If TRUE, notify on Twitter.

Examples

```
## Not run:
# get items by item ID
qiita_get_items(item_id = "7a78d897810446dd6a3b")

# get items by tag ID
qiita_get_items(tag_id = c("dplyr", "tidyr"), per_pages = 10L, page_limit = 1L)

# get items by user ID
qiita_get_items(user_id = "yutannihilation")

# Post an item. Note that the post is private by default.
# You should manually check if the post is valid before make it public.
item <- qiita_post_item(title = "test", body = "This is an example.")

# update the post
qiita_update_item(item$id, title = "test", body = "**This is a strong example!**")

# delete the post
qiita_delete_item(item$id)

## End(Not run)
```

`qiita_tag`*Qiita Tag API*

Description

Get, follow or unfollow tags via Qiita API.

Usage

```
qiita_get_tags(  
  tag_id = NULL,  
  user_id = NULL,  
  per_page = 100L,  
  page_offset = 0L,  
  page_limit = 1L  
)  
  
qiita_follow_tag(tag_id)  
  
qiita_unfollow_tag(tag_id)  
  
qiita_is_following_tag(tag_id)
```

Arguments

<code>tag_id</code>	Tag ID (e.g. "R", "dplyr").
<code>user_id</code>	User ID (e.g. "yutannihilation").
<code>per_page</code>	Number of items per one page.
<code>page_offset</code>	Page offset.
<code>page_limit</code>	Max number of pages to acquire.

Examples

```
## Not run:  
# get a tag by Tag IDs  
qiita_get_tags(tag_id = "R")  
  
# get tags by user ID  
qiita_get_tags(user_id = "yutannihilation")  
  
# follow a tag  
qiita_follow_tag("RStudio")  
  
# unfollow a tag  
qiita_unfollow_tag("RStudio")  
  
## End(Not run)
```

`qiita_user`*Qiita User API*

Description

Get, follow or unfollow users via Qiita API.

Usage

```
qiita_get_stockers(item_id, per_page = 100L, page_offset = 0L, page_limit = 1L)
```

```
qiita_get_users(user_id)
```

```
qiita_get_followees(  
  user_id,  
  per_page = 100L,  
  page_offset = 0L,  
  page_limit = 1L  
)
```

```
qiita_get_followers(  
  user_id,  
  per_page = 100L,  
  page_offset = 0L,  
  page_limit = 1L  
)
```

```
qiita_follow_user(user_id)
```

```
qiita_unfollow_user(user_id)
```

```
qiita_is_following_user(user_id)
```

```
qiita_get_authenticated_user()
```

Arguments

<code>item_id</code>	Item (article) ID.
<code>per_page</code>	Number of items per one page.
<code>page_offset</code>	Number of offset pages.
<code>page_limit</code>	Max number of pages to retrieve.
<code>user_id</code>	User ID (e.g. "yutannihilation").

Examples

```
## Not run:
# get a user by id
qiita_get_users("yutannihilation")

# follow a user
qiita_follow_user("user1")

# unfollow a user
qiita_unfollow_user("user1")

# get the current user
qiita_get_authenticated_user()

## End(Not run)
```

qiita_util

Generate Payload And Tag For Qiita API

Description

Generate Payload And Tag For Qiita API

Usage

```
qiita_util_tag(name, versions = NULL)
```

```
qiita_util_payload(
  body = NULL,
  title = NULL,
  tags = NULL,
  private = NULL,
  coediting = NULL,
  gist = NULL,
  tweet = NULL
)
```

Arguments

name	Tag name
versions	Versions (e.g. 3.1, >3.2).
body	Content body.
title	Title.
tags	Tags. Use <code>qiita_util_tag</code> to generate tag objects.
private	If TRUE, the post will be private.

<code>coediting</code>	If TRUE, the post will be editable by team members.
<code>gist</code>	If TRUE, post the code to Gist.
<code>tweet</code>	If TRUE, notify on Twitter.

Examples

```
qiita_util_tag(name = "R", versions = ">3.1")

qiita_util_payload(body = "foo",
  title = "test",
  tags = list(
    qiita_util_tag(name = "R", versions = ">3.1"),
    qiita_util_tag(name = "dplyr")
  ),
  private = TRUE)
```

Index

httr, 2

qiita_api, 2
qiita_auth, 2
qiita_comment, 3
qiita_delete_comment (qiita_comment), 3
qiita_delete_item (qiita_item), 4
qiita_follow_tag (qiita_tag), 6
qiita_follow_user (qiita_user), 7
qiita_get_authenticated_user
 (qiita_user), 7
qiita_get_comments (qiita_comment), 3
qiita_get_followees (qiita_user), 7
qiita_get_followers (qiita_user), 7
qiita_get_items (qiita_item), 4
qiita_get_stockers (qiita_user), 7
qiita_get_stocks (qiita_item), 4
qiita_get_tags (qiita_tag), 6
qiita_get_users (qiita_user), 7
qiita_is_following_tag (qiita_tag), 6
qiita_is_following_user (qiita_user), 7
qiita_is_stocked_item (qiita_item), 4
qiita_item, 4
qiita_post_comment (qiita_comment), 3
qiita_post_item (qiita_item), 4
qiita_set_accesstoken (qiita_auth), 2
qiita_stock_item (qiita_item), 4
qiita_tag, 6
qiita_unfollow_tag (qiita_tag), 6
qiita_unfollow_user (qiita_user), 7
qiita_unstock_item (qiita_item), 4
qiita_update_comment (qiita_comment), 3
qiita_update_item (qiita_item), 4
qiita_user, 7
qiita_util, 8
qiita_util_payload (qiita_util), 8
qiita_util_tag (qiita_util), 8

Startup, 2