Package ‘discgolf’

October 13, 2022

Title Discourse API Client

Description Client for the Discourse API. Discourse is an open source discussion forum platform (<https://www.discourse.org/>). It comes with 'RESTful' API access to an installation. This client requires that you are authorized to access a Discourse installation, either yours or another.

Version 0.2.0

License MIT + file LICENSE

URL https://github.com/sckott/discgolf

BugReports https://github.com/sckott/discgolf/issues

LazyData true

Imports crul (>= 0.4.0), jsonlite, xml2

Suggests testthat

RoxygenNote 6.0.1

NeedsCompilation no

Author Scott Chamberlain [aut, cre]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2018-01-03 22:34:26 UTC

R topics documented:

discgolf-package .................................................. 2
badges ............................................................... 2
categories ......................................................... 3
dg_search .......................................................... 5
disc_setup ......................................................... 6
e-mail .............................................................. 7
groups ............................................................. 7
notifications ...................................................... 8
posts ............................................................... 9
Description

R Client for Discourse API

Usage

badges(url = NULL, key = NULL, user = NULL, ...)
badges_user(username, url = NULL, key = NULL, user = NULL, ...)
badge_create(name, badge_type_id, description = NULL, allow_title = NULL,
multiple_grant = NULL, icon = NULL, listable = NULL,
target_posts = NULL, enabled = NULL, auto_revoke = NULL,
badge_grouping_id = NULL, show_posts = NULL, image = NULL,
long_description = NULL, url = NULL, key = NULL, user = NULL, ...)

Arguments

url Base url for a Discourse installation. See disc_setup
key Your api key on the Discourse installation. See disc_setup
user Your user name on the Discourse installation. See disc_setup
... Named parameters passed on to HttpClient
username A current user name. required
name (character) A badge name to be created. required
badge_type_id (integer) A badge type id. one of 1 (gold), 2 (silver), or 3 (bronze). required
description (character) A description(character) . optional
allow_title (logical) Allow a title. optional
categories

multiple_grant (character) multiple grant. optional
icon (character) icon. optional
listable (logical) listable. optional
target_posts (character). optional
enabled (logical) enabled or not. optional
auto_revoke (character) auto revoke. optional
badge_grouping_id (character) badge grouping id. optional
show_posts (logical) show posts. optional
image (character) an image. optional
long_description (character) long description. optional

Examples

## Not run:
# all badges
badges()

# specific badge by name
badges_user("sckott")

# create a badge
(res <- badge_create(name = "funnestbadge", badge_type_id = 3))

## End(Not run)

categories  Work with categories

Description

Work with categories

Usage

categories(url = NULL, key = NULL, user = NULL, ...)
category(category, url = NULL, key = NULL, user = NULL, ...)
category_latest_topics(category, url = NULL, key = NULL, user = NULL, ...)
category_top_topics(category, url = NULL, key = NULL, user = NULL, ...)
category_new_topics(category, url = NULL, key = NULL, user = NULL, ...)
categories

category_create(category, color, text_color, description = NULL,
permissions = NULL, parent_category = NULL, url = NULL, key = NULL,
user = NULL, ...)

Arguments

url          Base url for a Discourse installation. See disc_setup
key          Your api key on the Discourse installation. See disc_setup
user         Your user name on the Discourse installation. See disc_setup
...           Named parameters passed on to HttpClient
category      Category name. required
color         A color by name or hex string. optional
text_color    A color by name or hex string. optional
description   Description of the category. optional
permissions   Permissions - a list with the group name and permission_type which is an integer: 1 = Full, 2 = Create Post, 3 = Read Only. optional
parent_category x. optional

Details

Apparently there’s no ability to delete categories via the API.

Examples

## Not run:
# all categories
categories()

# a specific category
category("questions")
category("packages")

# latest topics for a category
category_latest_topics("packages")

# top topics for a category
category_top_topics("packages")

# new topics for a category
category_new_topics("packages")

# create a category
category_create("stuff3", "F7941D", "FFFFFF", "My new category")

## End(Not run)
Description

Search

Usage

dg_search(query = NULL, order = NULL, status = NULL, category = NULL,
          username = NULL, group = NULL, badge = NULL, in_ = NULL,
          posts_count = NULL, min_age = NULL, max_age = NULL, url = NULL,
          key = NULL, user = NULL, ...)

Arguments

query (character) Query terms. Required.
order (character) One of views, latest, likes
status (character) One of open, closed, archived, noreplies, or single_user
category (character) Category to search for
username (character) User name
group (character) Group name
badge (character) Badge name
in_ (character) One of likes, posted, watching, tracking, private, bookmarks, first
posts_count (integer) Number of posts per topic
min_age (integer) Minimum age
max_age (integer) Maximum age
url Base url for a Discourse installation. See disc_setup
key Your api key on the Discourse installation. See disc_setup
user Your user name on the Discourse installation. See disc_setup
... Named parameters passed on to HttpClient

Examples

## Not run:
dg_search(query = "poo")
dg_search(posts_count = 1)
dg_search(in_ = "posted")
dg_search(status = "open")

## End(Not run)
**Description**

Discgolf setup

**How to setup**

You can set your url, username, and api key either using R options or environment variables. I recommend using environment variables since they are more general to any programming language, and you can easily set secure env vars on a server or e.g., if you’re doing continuous integration on Travis-CI (or elsewhere).

**URL**

The base URL for the Discourse instance, e.g., https://meta.discourse.org

Use:

- Env var name: DISCOURSE_URL
- R option name: discourse_url

**Username**

The user name you have registered on the Discourse instance you want to use.

Use:

- Env var name: DISCOURSE_USERNAME
- R option name: discourse_username

**API key**

The API key on the Discourse instance you want to use. This is not your password you used to login to the instance. If you’re the admin, you can get an API key by going the dashboard at base_url/admin, then the API tab base_url/admin/api, then generate a key, or copy the one already there.

Use:

- Env var name: DISCOURSE_API_KEY
- R option name: discourse_api_key
email

Description
Email

Usage
email_settings(url = NULL, key = NULL, user = NULL, ...)
list_email(filter, url = NULL, key = NULL, user = NULL, ...)

Arguments
url Base url for a Discourse installation. See disc_setup
key Your api key on the Discourse installation. See disc_setup
user Your user name on the Discourse installation. See disc_setup
... Named parameters passed on to HttpClient
filter An email filter, one of received, sent, skipped, rejected

Examples
## Not run:
# email settings
email_settings()

# list settings for an email address
list_email("received")

## End(Not run)

groups Work with groups

Description
Work with groups

Usage
groups(url = NULL, key = NULL, user = NULL, ...)
group_members(name, url = NULL, key = NULL, user = NULL, ...)
group_create(name, url = NULL, key = NULL, user = NULL, ...)
group_delete(id, url = NULL, key = NULL, user = NULL, ...)
### Arguments

- **url**: Base url for a Discourse installation. See [disc_setup](#)
- **key**: Your api key on the Discourse installation. See [disc_setup](#)
- **user**: Your user name on the Discourse installation. See [disc_setup](#)
- **...**: Named parameters passed on to [HttpClient](#)
- **name**: (character) A group name. required
- **id**: (numeric) A group id. required

### Examples

```r
## Not run:
# all groups
groups()

# specific group by name
group_members("admins")
group_members("moderators")
group_members("trust_level_3")

# create a group
(res <- group_create("group_testing2"))

# delete a group
group_delete(res$basic_group$id)

## End(Not run)
```

### Description

Notifications data

### Usage

```r
notifications(url = NULL, key = NULL, user = NULL, ...)
```

### Arguments

- **url**: Base url for a Discourse installation. See [disc_setup](#)
- **key**: Your api key on the Discourse installation. See [disc_setup](#)
- **user**: Your user name on the Discourse installation. See [disc_setup](#)
- **...**: Named parameters passed on to [HttpClient](#)
posts

Examples

## Not run:

`notifications()`

## End(Not run)

---

posts   Work with posts

Description

Work with posts

Usage

```r
post_get(post_id, url = NULL, key = NULL, user = NULL, ...)
post_create(topic_id, text, url = NULL, key = NULL, user = NULL, ...)
post_wikify(post_id, url = NULL, key = NULL, user = NULL, ...)
```

Arguments

- `post_id`: A post id
- `url`: Base url for a Discourse installation. See `disc_setup`
- `key`: Your api key on the Discourse installation. See `disc_setup`
- `user`: Your user name on the Discourse installation. See `disc_setup`
- `...`: Named parameters passed on to `HttpClient`
- `topic_id`: A topic id
- `text`: Text to put in post

Examples

## Not run:

# get a post
post_get(90)
post_get(120)
post_get(130)
post_get(155)
cat(post_get(155)$raw)

# create post, create topic first
z <- topic_create("The problems with blue skies",
                 text = "just saying these things cause I like the sky very much")
zz <- post_create(topic_id = z$topic_id,
                 text = "There isn't a problem is there?")
private_messages

Description

Private messages

Usage

private_messages(username, url = NULL, key = NULL, user = NULL, ...)

Arguments

username A username. required.
url Base url for a Discourse installation. See disc_setup
key Your api key on the Discourse installation. See disc_setup
user Your user name on the Discourse installation. See disc_setup
... Named parameters passed on to HttpClient

Examples

## Not run:
private_messages(username = "sckott")

## End(Not run)
### tag

**Tag data**

**Description**

Tag data

**Usage**

```r
tag(tag, url = NULL, key = NULL, user = NULL, ...)
```

**Arguments**

- `tag`: A tag name
- `url`: Base url for a Discourse installation. See `disc_setup`
- `key`: Your api key on the Discourse installation. See `disc_setup`
- `user`: Your user name on the Discourse installation. See `disc_setup`
- `...`: Named parameters passed on to `HttpClient`

**Examples**

```r
## Not run:
tag('r')
tag('geospatial')
## End(Not run)
```

### topics

**Work with topics**

**Description**

Work with topics

**Usage**

```r
topics_latest(url = NULL, key = NULL, user = NULL, ...)
topics_new(url = NULL, key = NULL, user = NULL, ...)
topics_by(id, url = NULL, key = NULL, user = NULL, ...)
topic(id, url = NULL, key = NULL, user = NULL, ...)
topic_create(title, text, category = NULL, url = NULL, key = NULL,
```
user = NULL, ...

```
topic_delete(id, url = NULL, key = NULL, user = NULL, ...)
```

```
topic_rename(id, title, url = NULL, key = NULL, user = NULL, ...)
```

### Arguments

- **url**
  Base url for a Discourse installation. See `disc_setup`
- **key**
  Your api key on the Discourse installation. See `disc_setup`
- **user**
  Your user name on the Discourse installation. See `disc_setup`
- ...  
  Named parameters passed on to `HttpClient`
- **id**
  A user identifier, or a topic identifier. See examples.
- **title**
  Title of the new topic
- **text**
  Raw text, can include markdown and will be rendered.
- **category**
  Category to apply to topic, if any. Default: none

### Examples

```r
## Not run:
topics_latest()
topics_new()
topics_by("cboettig")
topic(8)

topics_latest(verbos = TRUE)

# Create topic
## simple
topic_create(title = "testing from discgolf - 10", text = "testing from discgolf, hello world! hopefully this works")

## more complicated
text <- '
## this time with markdown
```
```
print("hello world")
#> [1] "hello world"
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
users

Work with users

Description

Work with users

Usage

user(username, url = NULL, key = NULL, user = NULL, ...)

users_list(type, url = NULL, key = NULL, user = NULL, ...)

user_create(name, email, username, password, url = NULL, key = NULL, user = NULL, ...)

user_activate(user_id, url = NULL, key = NULL, user = NULL, ...)

user_delete(user_id, url = NULL, key = NULL, user = NULL, ...)

user_update_email(username, email, url = NULL, key = NULL, user = NULL, ...)

user_update_username(username, new_username, url = NULL, key = NULL, user = NULL, ...)

```r
>>> Valiant 18.1 6 25 105 2.76 3.46 20.22 1 0 3 1
```

topic_create(title = "testing from discgolf - 23", text = text)

# delete a topic

topic_delete(242)

# rename a topic - DOESN'T WORK YET

## create a topic

# res <- topic_create(title = "testing from discgolf - 5",
# text = "testing from discgolf, again")

## rename

# res$topic_slug

# topic_rename(res$topic_id, title = "new title")

# topic(res$topic_id)$title

## cleanup

# topic_delete(res$topic_id)

## End(Not run)
Arguments

- **username**: A user name
- **url**: Base url for a Discourse installation. See `disc_setup`
- **key**: Your api key on the Discourse installation. See `disc_setup`
- **user**: Your user name on the Discourse installation. See `disc_setup`
- **type**: A type of user, one of active, staff, new, suspended, blocked, or suspect
- **name**: a name
- **email**: an email address
- **password**: a password
- **user_id**: a user id
- **new_username**: a username

users_list

Note that there is no paging, so if you have more than 100 users, you only get the first 100. :sad

Examples

```r
## Not run:
# list a user
user('sckott')
user('cboettig')

# list users
users_list('staff')
users_list('new')

# create a user
(x <- user_create("jane doe", "janie@doe.com", "jane_doe",
"FTh6bmoPaaWr1ISflDM6wy")

# activate a user
user_activate(x$user_id)

# update email address
user_update_email(username = 'jane_doe', email = 'jane2@doe.com')

# update user name
user_update_username('jane_doe', 'jane_doe5')

# delete a user
user_delete(x$user_id)

## End(Not run)
```
Index

* package
  discgolf-package, 2
badge_create (badges), 2
badges, 2
badges_user (badges), 2
categories, 3
category (categories), 3
category_create (categories), 3
category_latest_topics (categories), 3
category_new_topics (categories), 3
category_top_topics (categories), 3
dg_search, 5
disc_setup, 2, 4, 5, 6, 7–12, 14
discgolf (discgolf-package), 2
discgolf-package, 2
e-mail, 7
e-mail_settings (email), 7
group_create (groups), 7
group_delete (groups), 7
group_members (groups), 7
groups, 7
HttpClient, 2, 4, 5, 7–12, 14
list_email (email), 7
notifications, 8
post_create (posts), 9
post_get (posts), 9
post_wikify (posts), 9
posts, 9
private_messages, 10
tag, 11
topic (topics), 11
topic_create (topics), 11
topic_delete (topics), 11
topic_rename (topics), 11
topics, 11
topics_by (topics), 11
topics_latest (topics), 11
topics_new (topics), 11
user (users), 13
user_activate (users), 13
user_create (users), 13
user_delete (users), 13
user_update_email (users), 13
user_update_username (users), 13
users, 13
users_list (users), 13