Package ‘ghee’

March 11, 2021

Title Provides a Lightweight Interface for 'GitHub' through R
Version 0.1.0
Description Provides a user friendly wrapper for the 'gh' package facilitating easy access to the REST API for 'GitHub'. Includes support for common tasks such as creating and commenting on issues, inviting collaborators, and more.
License MIT + file LICENSE
Encoding UTF-8
Language es
LazyData true
RoxygenNote 7.1.1.9000
URL https://github.com/jdtrat/ghee
BugReports https://github.com/jdtrat/ghee/issues
Imports gh, jsonlite, utils
NeedsCompilation no
Author Jonathan Trattner [aut, cre] (<https://orcid.org/0000-0002-1097-7603>)
Maintainer Jonathan Trattner <jdt@jdtrat.com>
Repository CRAN
Date/Publication 2021-03-11 14:30:05 UTC

R topics documented:

  gh_collab_check .................................................. 2
  gh_collab_invite .................................................. 3
  gh_collab_list .................................................... 3
  gh_collab_pending ............................................... 4
  gh_collab_remove ................................................ 5
  gh_collab_uninvite ............................................. 5
  gh_issue_assign .................................................. 6
  gh_issue_comment ............................................... 7
  gh_issue_list .................................................... 7
gh_collab_check

Check if a user is a collaborator

Description

Check if a user is a collaborator

Usage

gh_collab_check(path, collaborator, ..., messages = TRUE)

Arguments

path
GitHub repo path of the form "username/repo". If not provided, will default to current repo.
collaborator
The collaborator for whom to call this function.
...
Additional parameters, see https://docs.github.com/en/rest
messages
Logical: Should a message indicating the status of the function be printed? TRUE by default.

Value

TRUE if the individual is a collaborator on the repo; FALSE otherwise.

See Also

gh_collab_list() for listing all current repository collaborators.
Other Collaborations: gh_collab_invite(), gh_collab_list(), gh_collab_pending(), gh_collab_remove(), gh_collab_uninvite()
**`gh_collab_invite`**

*Invite a collaborator to a GitHub repo*

**Description**
Invite a collaborator to a GitHub repo

**Usage**
```
gh_collab_invite(path, collaborator, ...)
```

**Arguments**
- **path**: GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- **collaborator**: The collaborator for whom to call this function.
- **...**: Additional parameters, see https://docs.github.com/en/rest

**Value**
NA; used for side effects to invite a collaborator to the specified GitHub repository.

**See Also**
Other Collaborations: `gh_collab_check()`, `gh_collab_list()`, `gh_collab_pending()`, `gh_collab_remove()`, `gh_collab_uninvite()`

---

**`gh_collab_list`**

*List current collaborators for a GitHub repo*

**Description**
List current collaborators for a GitHub repo

**Usage**
```
gh_collab_list(path, ...)
```

**Arguments**
- **path**: GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- **...**: Additional parameters, see https://docs.github.com/en/rest
Value

A data frame that contains the collaborators for the specified repository.

See Also

- `gh_collab_check()` for checking whether a specific user is a collaborator.
- Other Collaborations: `gh_collab_check()`, `gh_collab_invite()`, `gh_collab_pending()`, `gh_collab_remove()`, `gh_collab_uninvite()`

---

**gh_collab_pending**

_List pending invitations for a repo_

Description

List pending invitations for a repo

Usage

```
gh_collab_pending(path, ...)
```

Arguments

- **path**
  - GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- **...**
  - Additional parameters, see [https://docs.github.com/en/rest](https://docs.github.com/en/rest)

Value

A dataframe listing the usernames of pending invitees (or dataframe of length 0 if no pending invitees).

See Also

- Other Collaborations: `gh_collab_check()`, `gh_collab_invite()`, `gh_collab_list()`, `gh_collab_remove()`, `gh_collab_uninvite()`
**gh_collab_remove**  
*Remove a collaborator from a repository*

**Description**

This will remove a collaborator from a repository and confirm the individual is no longer a member.

**Usage**

```r
gh_collab_remove(path, collaborator, ..., messages = TRUE)
```

**Arguments**

- `path`: GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- `collaborator`: The collaborator for whom to call this function.
- `...`: Additional parameters, see [https://docs.github.com/en/rest](https://docs.github.com/en/rest)
- `messages`: Logical: Should a message indicating the status of the function be printed? TRUE by default.

**Value**

NA; called for its side effects.

**See Also**

Other Collaborations: `gh_collab_check()`, `gh_collab_invite()`, `gh_collab_list()`, `gh_collab_pending()`, `gh_collab_uninvite()`

---

**gh_collab_uninvite**  
*Uninvite a collaborator to a GitHub repo*

**Description**

This only works if the individual’s invite is pending.

**Usage**

```r
gh_collab_uninvite(path, collaborator, ...)
```

**Arguments**

- `path`: GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- `collaborator`: The collaborator for whom to call this function.
- `...`: Additional parameters, see [https://docs.github.com/en/rest](https://docs.github.com/en/rest)
Value
NA; called for side effects to uninvite a collaborator to the specified GitHub repository.

See Also
Other Collaborations: `gh_collab_check()`, `gh_collab_invite()`, `gh_collab_list()`, `gh_collab_pending()`, `gh_collab_remove()`

---

### gh_issue_assign

**Assign people to a GitHub issue**

**Description**
Assign people to a GitHub issue

**Usage**

gh_issue_assign(path, issue_number, collaborator, ...)

**Arguments**

- **path**
  GitHub repo path of the form "username/repo". If not provided, will default to current repo.

- **issue_number**
The issue number on which to comment. Can be determined online or with `gh_issue_list`

- **collaborator**
The collaborator for whom to call this function.

- **...**
  Additional parameters, see https://docs.github.com/en/rest

**Value**
NA; called for side effects

**See Also**
Other Issues: `gh_issue_comment()`, `gh_issue_list()`, `gh_issue_mention()`, `gh_issue_new()`
**gh_issue_comment**  
*Comment on a GitHub issue*

**Description**

Comment on a GitHub issue

**Usage**

```r
gh_issue_comment(path, issue_number, body, ...)
```

**Arguments**

- **path**: GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- **issue_number**: The issue number on which to comment. Can be determined online or with `gh_issue_list`
- **body**: The issue's body text
- **...**: Additional parameters, see [https://docs.github.com/en/rest](https://docs.github.com/en/rest)

**Value**

NA; called for side effects.

**See Also**

Other Issues: `gh_issue_assign()`, `gh_issue_list()`, `gh_issue_mention()`, `gh_issue_new()`

---

**gh_issue_list**  
*List issues for a GitHub Repo*

**Description**

List issues for a GitHub Repo

**Usage**

```r
gh_issue_list(path, ...)
```

**Arguments**

- **path**: GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- **...**: Additional parameters, see [https://docs.github.com/en/rest](https://docs.github.com/en/rest)
**Value**

A list of issues for the specified GitHub repo

**See Also**

Other Issues: gh_issue_assign(), gh_issue_comment(), gh_issue_mention(), gh_issue_new()

---

**gh_issue_mention**  
*Check to see if a person was mentioned in a GitHub issue*

**Description**

Check to see if a person was mentioned in a GitHub issue

**Usage**

gh_issue_mention(path, collaborator, ...)

**Arguments**

- **path**  
  GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- **collaborator**  
  The collaborator for whom to call this function.
- **...**  
  Additional parameters, see https://docs.github.com/en/rest

**Value**

TRUE if the collaborator was mentioned in any issues; FALSE otherwise

**See Also**

Other Issues: gh_issue_assign(), gh_issue_comment(), gh_issue_list(), gh_issue_new()

---

**gh_issue_new**  
*Create an issue on a github repo*

**Description**

Create an issue on a github repo

**Usage**

gh_issue_new(path, title, body, ...)
gh_repos_create

Arguments

- **path**: GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- **title**: The issue's title
- **body**: The issue's body text
- **...**: Additional parameters, see `https://docs.github.com/en/rest`

Value

NA; used for side effects to create a new issue on the specified GitHub repository.

See Also

Other Issues: `gh_issue_assign()`, `gh_issue_comment()`, `gh_issue_list()`, `gh_issue_mention()`

---

gh_repos_create Create a new repository

Description

Create a new repository

Usage

gh_repos_create(path, private = TRUE, description = NULL, ...)

Arguments

- **path**: GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- **private**: Will this repository be private? TRUE/FALSE; default is TRUE.
- **description**: (Optional) A short description of the repository.
- **...**: Additional parameters, see `https://docs.github.com/en/rest`

Value

NA; used for side effects to create a repository on GitHub.

See Also

Other Repositories: `gh_repos_delete_internal()`, `gh_repos_delete()`, `gh_repos_list()`, `gh_repos_mutate()`
**Delete a GitHub repository**

**Description**

*Use with caution!* By default the github token created with `usethis::create_github_token()` does not allow this functionality. You must create your own PAT that has the appropriate permissions to delete repositories.

**Usage**

```r
gh_repos_delete(path, ...)
```

**Arguments**

- `path` GitHub repo path of the form "username/repo". If not provided, will default to current repo.
- `...` Additional parameters, see [https://docs.github.com/en/rest](https://docs.github.com/en/rest)

**Value**

NA; used for side effects to delete a repository from GitHub.

**See Also**

Other Repositories: `gh_repos_create()`, `gh_repos_delete_internal()`, `gh_repos_list()`, `gh_repos_mutate()`

---

**List a user's github repository**

**Description**

List a user’s github repository

**Usage**

```r
gh_repos_list(user, ...)
```

**Arguments**

- `user` The GitHub user
- `...` Additional parameters, see [https://docs.github.com/en/rest](https://docs.github.com/en/rest)

**Value**

Repositories for the specified user.
gh_repos_mutate

See Also

Other Repositories: gh_repos_create(), gh_repos_delete_internal(), gh_repos_delete(), gh_repos_mutate()

Examples

gh_repos_list("jdtrat")

gh_repos_mutate Change repository features

Description

This function allows you to update a repository’s features. You can easily change the repository’s name, privacy settings, and more.

Usage

gh_repos_mutate(path, ...)

Arguments

path GitHub repo path of the form "username/repo". If not provided, will default to current repo.

... Additional parameters, see https://docs.github.com/en/rest

Value

NA; called for side effects to manipulate GitHub repositories.

See Also

Other Repositories: gh_repos_create(), gh_repos_delete_internal(), gh_repos_delete(), gh_repos_list()

Examples

## Not run:

# Create a Private Repo
gh_repos_create(path = "jdtrat/ghee_test", private = TRUE, description = "A test repository.")

# Disable Issues
gh_repos_mutate(path = "jdtrat/ghee_test", has_issues = FALSE)

# Update Description
gh_repos_mutate(path = "jdtrat/ghee_test", description = "A test repository for {ghee}.")
# Remove Description
gh_repos_mutate("jdtrat/friend", description = NA)

# Change Privacy Settings
gh_repos_mutate(path = "jdtrat/ghee_test", private = FALSE)

# Change Repo Name
gh_repos_mutate(path = "jdtrat/ghee_test", name = "ghee_testing")

# Change Repo Name Back
# Note the path argument reflects the name change
gh_repos_mutate(path = "jdtrat/ghee_testing", name = "ghee_test")

# Delete Repo
# Note this requires a special GitHub Token and should be used with caution.
gh_repos_delete("jdtrat/ghee_test")

## End(Not run)
Index

* Collaborations
  gh_collab_check, 2
  gh_collab_invite, 3
  gh_collab_list, 3
  gh_collab_pending, 4
  gh_collab_remove, 5
  gh_collab_uninvite, 5

* Issues
  gh_issue_assign, 6
  gh_issue_comment, 7
  gh_issue_list, 7
  gh_issue_mention, 8
  gh_issue_new, 8

* Repositories
  gh_repos_create, 9
  gh_repos_delete, 10
  gh_repos_list, 10
  gh_repos_mutate, 11

  gh_collab_check, 2, 3–6
  gh_collab_check(), 4
  gh_collab_invite, 2, 3, 4–6
  gh_collab_list, 2, 3, 4–6
  gh_collab_list(), 2
  gh_collab_pending, 2–4, 4, 5, 6
  gh_collab_remove, 2–4, 5, 6
  gh_collab_uninvite, 2–5, 5
  gh_issue_assign, 6, 7–9
  gh_issue_comment, 6, 7, 8, 9
  gh_issue_list, 6, 7, 8, 9
  gh_issue_mention, 6–8, 8, 9
  gh_issue_new, 6–8, 8
  gh_repos_create, 9, 10, 11
  gh_repos_delete, 9, 10, 11
  gh_repos_delete_internal, 9–11
  gh_repos_list, 9, 10, 10, 11
  gh_repos_mutate, 9–11, 11

  usethis::create_github_token(), 10