

Package ‘gh’

July 16, 2017

Title 'GitHub' 'API'

Version 1.0.1

Description Minimal client to access the 'GitHub' 'API'.

License MIT + file LICENSE

LazyData true

URL <https://github.com/r-lib/gh#readme>

BugReports <https://github.com/r-lib/gh/issues>

Suggests covr, pingr, testthat

Imports ini, jsonlite, httr

RoxygenNote 6.0.1.9000

Encoding UTF-8

NeedsCompilation no

Author Gábor Csárdi [cre, ctb],
Jennifer Bryan [aut],
Hadley Wickham [aut]

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Repository CRAN

Date/Publication 2017-07-16 20:28:25 UTC

R topics documented:

gh	2
gh_next	4
gh_tree_remote	5
gh_whoami	5
print.gh_response	7

Index	8
--------------	----------

Description

Minimal wrapper to access GitHub's API.

This is an extremely minimal client. You need to know the API to be able to use this client. All this function does is:

- Try to substitute each listed parameter into endpoint, using the `:parameter` notation.
- If a GET request (the default), then add all other listed parameters as query parameters.
- If not a GET request, then send the other parameters in the request body, as JSON.
- Convert the response to an R list using `jsonlite::fromJSON`.

Usage

```
gh(endpoint, ..., .token = NULL, .api_url = NULL, .method = "GET",
    .limit = NULL, .send_headers = NULL)
```

Arguments

<code>endpoint</code>	GitHub API endpoint. Must be one of the following forms: <ul style="list-style-type: none"> • "METHOD path", e.g. "GET /rate_limit" • "path", e.g. "/rate_limit". • "METHOD url", e.g. "GET https://api.github.com/rate_limit" • "url", e.g. "https://api.github.com/rate_limit". If the method is not supplied, will use <code>.method</code> , which defaults to GET.
<code>...</code>	Name-value pairs giving API parameters. Will be matched into <code>url</code> placeholders, sent as query parameters in GET requests, and in the JSON body of POST requests.
<code>.token</code>	Authentication token.
<code>.api_url</code>	Github API url (default: https://api.github.com). Used if endpoint just contains a path.
<code>.method</code>	HTTP method to use if not explicitly supplied in the endpoint.
<code>.limit</code>	Number of records to return. This can be used instead of manual pagination. By default it is NULL, which means that the defaults of the GitHub API are used. You can set it to a number to request more (or less) records, and also to <code>Inf</code> to request all records. Note, that if you request many records, then multiple GitHub API calls are used to get them, and this can take a potentially long time.
<code>.send_headers</code>	Named character vector of header field values (excepting <code>Authorization</code> , which is handled via <code>.token</code>). This can be used to override or augment the defaults, which are as follows: the <code>Accept</code> field defaults to <code>"application/vnd.github.v3+json"</code> and the <code>User-Agent</code> field defaults to <code>"https://github.com/r-lib/gh"</code> . This can be used to, e.g., provide a custom media type, in order to access a preview feature of the API.

Value

Answer from the API as a `gh_response` object, which is also a list. Failed requests will generate an R error.

See Also

[gh_whoami\(\)](#) for details on GitHub API token management.

Examples

```
## Not run:
## Repositories of a user, these are equivalent
gh("/users/hadley/repos")
gh("/users/:username/repos", username = "hadley")

## Starred repositories of a user
gh("/users/hadley/starred")
gh("/users/:username/starred", username = "hadley")

## Create a repository, needs a token in GITHUB_PAT (or GITHUB_TOKEN)
## environment variable
gh("POST /user/repos", name = "foobar")

## Issues of a repository
gh("/repos/hadley/dplyr/issues")
gh("/repos/:owner/:repo/issues", owner = "hadley", repo = "dplyr")

## Automatic pagination
users <- gh("/users", .limit = 50)
length(users)

## Access developer preview of Licenses API (in preview as of 2015-09-24)
gh("/licenses") # error code 415
gh("/licenses",
  .send_headers = c("Accept" = "application/vnd.github.drax-preview+json"))

## Access Github Enterprise API
gh("/user/repos", type = "public", .api_url = "https://github.foobar.edu/api/v3")

## Use I() to force body part to be sent as an array, even if length 1
## This works whether assignees has length 1 or > 1
assignees <- "gh_user"
assignees <- c("gh_user1", "gh_user2")
gh("PATCH /repos/OWNER/REPO/issues/1", assignees = I(assignees))

## End(Not run)
```

`gh_next`*Get the next, previous, first or last page of results*

Description

Get the next, previous, first or last page of results

Usage

```
gh_next(gh_response)
```

```
gh_prev(gh_response)
```

```
gh_first(gh_response)
```

```
gh_last(gh_response)
```

Arguments

`gh_response` An object returned by a `gh()` call.

Details

Note that these are not always defined. E.g. if the first page was queried (the default), then there are no first and previous pages defined. If there is no next page, then there is no next page defined, etc.

If the requested page does not exist, an error is thrown.

Value

Answer from the API.

Examples

```
## Not run:  
x <- gh("/users")  
sapply(x, "[[", "login")  
x2 <- gh_next(x)  
sapply(x2, "[[", "login")  
  
## End(Not run)
```

gh_tree_remote	<i>Find the GitHub remote associated with a path</i>
----------------	--

Description

This is handy helper if you want to make gh requests related to the current project.

Usage

```
gh_tree_remote(path = ".")
```

Arguments

path	Path that is contained within a git repo.
------	---

Value

If the repo has a github remote, a list containing username and repo. Otherwise, an error.

Examples

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

gh_whoami	<i>Info on current GitHub user and token</i>
-----------	--

Description

Reports wallet name, GitHub login, and GitHub URL for the current authenticated user, the first bit of the token, and the associated scopes.

Usage

```
gh_whoami(.token = NULL, .api_url = NULL, .send_headers = NULL)
```

Arguments

.token	Authentication token.
.api_url	Github API url (default: https://api.github.com). Used if endpoint just contains a path.

`.send_headers` Named character vector of header field values (excepting Authorization, which is handled via `.token`). This can be used to override or augment the defaults, which are as follows: the Accept field defaults to "application/vnd.github.v3+json" and the User-Agent field defaults to "https://github.com/r-lib/gh". This can be used to, e.g., provide a custom media type, in order to access a preview feature of the API.

Details

Get a personal access token for the GitHub API from <https://github.com/settings/tokens> and select the scopes necessary for your planned tasks. The repo scope, for example, is one many are likely to need. The token itself is a string of 40 letters and digits. You can store it any way you like and provide explicitly via the `.token` argument to `gh()`.

However, many prefer to define an environment variable GITHUB_PAT (or GITHUB_TOKEN) with this value in their `.Renvi` file. Add a line that looks like this, substituting your PAT:

```
GITHUB_PAT=8c70fd8419398999c9ac5bacf3192882193cadf2
```

Put a line break at the end! If you're using an editor that shows line numbers, there should be (at least) two lines, where the second one is empty. Restart R for this to take effect. Call `gh_whoami()` to confirm success.

To get complete information on the authenticated user, call `gh("/user")`.

For token management via API (versus the browser), use the [OAuth Authorizations API](#). This API requires Basic Authentication using your username and password, not tokens, and is outside the scope of the `gh` package.

Value

A `gh_response` object, which is also a list.

Examples

```
## Not run:
gh_whoami()

## explicit token + use with GitHub Enterprise
gh_whoami(.token = "8c70fd8419398999c9ac5bacf3192882193cadf2",
          .api_url = "https://github.foobar.edu/api/v3")

## End(Not run)
```

`print.gh_response` *Print the result of a GitHub API call*

Description

Print the result of a GitHub API call

Usage

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

Arguments

<code>x</code>	The result object.
<code>...</code>	Ignored.

Value

The JSON result.

Index

gh, [2](#), [6](#)
gh-package (gh), [2](#)
gh_first (gh_next), [4](#)
gh_last (gh_next), [4](#)
gh_next, [4](#)
gh_prev (gh_next), [4](#)
gh_tree_remote, [5](#)
gh_whoami, [3](#), [5](#)

print.gh_response, [7](#)