

Package ‘gitlabr’

February 9, 2016

Title Access to the Gitlab API

Version 0.6.4

Description Provides R functions to access the API of the project and repository management web application gitlab. For many common tasks (repository file access, issue assignment and status, commenting) convenience wrappers are provided, and in addition the full API can be used by specifying request locations. Gitlab is open-source software and can be self-hosted or used on gitlab.com.

Depends R (>= 3.1.2), magrittr

Imports dplyr (>= 0.4.3), stringr, base64enc, httr (>= 1.1.0), functional

Suggests devtools (>= 1.8.0), roxygen2, testthat, R.rsp, knitr

URL <http://blog.points-of-interest.cc/post/gitlabr/>
<http://doc.gitlab.com/ce/api/>

BugReports <http://gitlab.points-of-interest.cc/points-of-interest/gitlabr/issues/>

VignetteBuilder R.rsp

License GPL (>= 3)

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Author Jirka Lewandowski [aut, cre]

Maintainer Jirka Lewandowski <jirka.lewandowski@wzb.eu>

Repository CRAN

Date/Publication 2016-02-09 20:53:59

R topics documented:

archive	2
create_merge_request	3
file_exists	3
get_comments	4
get_commits	5
get_issues	5
get_project_id	6
gitlab	6
gitlabr	7
gitlab_connection	7
iff	9
list_branches	9
list_projects	10
new_issue	10
pipe_into	11
prefix	11
proj_req	12
push_file	12
repository	13
set_gitlab_connection	13
to_issue_id	14

Index **15**

archive	<i>Get zip archive of a specific repository</i>
---------	---

Description

Get zip archive of a specific repository

Usage

```
archive(project, save_to_file = tempfile(fileext = ".zip"), ...)
```

Arguments

project	Project name or id
save_to_file	path where to save archive; if this is NULL, the archive itself is returned as a raw vector
...	further parameters passed on to gitlab API call, may include parameter sha for specifying a commit hash

Value

if save_to_file is NULL, a raw vector of the archive, else the path to the saved archived file

create_merge_request *Create a merge request*

Description

Create a merge request

Usage

```
create_merge_request(project, source_branch, target_branch = "master", title,
                    description, verb = http::POST, ...)
```

Arguments

project	name or id of project (not repository!)
source_branch	name of branch to be merged
target_branch	name of branch into which to merge
title	title of the merge request
description	description text for the merge request
verb	is ignored, will always be forced to match the action the function name indicates
...	passed on to gitlab . Might contain more fields documented in gitlab API doc.

file_exists *For file_exists dots are passed on to [list_files](#) and gitlab API call*

Description

For file_exists dots are passed on to [list_files](#) and gitlab API call
 Get a file from a gitlab repository

Usage

```
file_exists(project, file_path, ...)

get_file(project, file_path, ref = "master", to_char = TRUE, ...)
```

Arguments

project	name or id of project
file_path	path to file
...	passed on to gitlab
ref	name of ref (commit branch or tag)
to_char	flag if output should be converted to char; otherwise it is of class raw

`get_comments`*Get the comments of a commit or issue*

Description

Get the comments of a commit or issue

Usage

```
get_comments(object_type = "issue", id, note_id = c(), project, ...)
```

```
get_issue_comments(...)
```

```
get_commit_comments(...)
```

```
comment_commit(project, id, text, ...)
```

```
comment_issue(project, id, text, ...)
```

```
edit_comment(object_type, text, ...)
```

```
edit_issue_comment(...)
```

```
edit_commit_comment(...)
```

Arguments

<code>object_type</code>	one of "issue" or "commit". Snippets and merge_requests are not implemented yet.
<code>id</code>	id of object: (project-wide) issue_id or commit sha
<code>note_id</code>	id of note
<code>project</code>	project name or id
<code>...</code>	passed on to gitlab API call. See Details.
<code>text</code>	Text of comment/note to add or edit (translates to gitlab API note/body respectively)

Details

For `comment_commit ...` might also contain `path`, `line` and `line_type` (old or new) to attach the comment to a specific in a file. See <http://doc.gitlab.com/ce/api/commits.html>

get_commits *Get commits and diff from a project repository*

Description

Get commits and diff from a project repository

Usage

```
get_commits(project, commit_sha = c(), ...)
```

```
get_diff(project, commit_sha, ...)
```

Arguments

project	project name or id
commit_sha	if not null, get only the commit with the specific hash; for get_diff this must be specified
...	passed on to gitlab API call, may contain ref_name for specifying a branch or tag to list commits of

get_issues *Get issues of a project or user*

Description

Get issues of a project or user

Usage

```
get_issues(project = NULL, issue_id = NULL, verb = httr::GET, ...)
```

```
get_issue(issue_id, project, ...)
```

Arguments

project	project name or id, may be null for all issues created by user
issue_id	optional issue id (projectwide iid, not gitlab API id)
verb	ignored; all calls with this function will have gitlab 's default verb httr::GET
...	further parameters passed on to gitlab , may be state, labels, issue id, ...

Details

get_issue provides a wrapper with swapped arguments for convenience, esp. when using a project connection

get_project_id	<i>Get a project id by name</i>
----------------	---------------------------------

Description

Get a project id by name

Usage

```
get_project_id(project_name, verb = httr::GET, auto_format = TRUE, ...)
```

Arguments

project_name	project name
verb	ignored; all calls with this function will have gitlab 's default verb <code>httr::GET</code>
auto_format	ignored
...	passed on to gitlab

gitlab	<i>Request Gitlab API</i>
--------	---------------------------

Description

Request Gitlab API

Usage

```
gitlab(req, api_root, verb = httr::GET, auto_format = TRUE, debug = FALSE,
  gitlab_con = "default", page = "all", enforce_api_root = TRUE,
  argname_verb = if (identical(verb, httr::GET) | identical(verb,
  httr::DELETE)) { "query" } else { "body" }, ...)
```

Arguments

req	vector of characters that represents the call (e.g. <code>c("projects", project_id, "events")</code>)
api_root	URL where the gitlab API to request resides (e.g. <code>https://gitlab.myserver.com/api/v3/</code>)
verb	http verb to use for request in form of one of the <code>httr</code> functions GET , PUT , POST , DELETE
auto_format	whether to format the returned object automatically to a flat data.frame
debug	if TRUE API URL and query will be printed, defaults to FALSE
gitlab_con	function to use for issuing API requests (e.g. as returned by gitlab_connection)
page	number of page of API response to get; if "all" (default), all pages are queried successively and combined.

enforce_api_root	if multiple pages are requested, the API root URL is ensured to be the same as in the original call for all calls using the "next page" URL returned by gitlab. This makes sense for security and in cases where gitlab is behind a reverse proxy and ignorant about its URL from external.
argname_verb	name of the argument of the verb that fields and information are passed on to
...	named parameters to pass on to gitlab API (technically: modifies query parameters of request URL), may include private_token and all other parameters as documented for the Gitlab API

gitlabr *Interface to gitlab API on high and low levels*

Description

gitlabr R package

Details

Interface to gitlab API on high and low levels

Package:	gitlabr
Type:	Package
License:	GPL (>= 3)
LazyLoad:	yes

Author(s)

Jirka Lewandowski <jirka.lewandowski@wzb.eu>

References

<http://blog.points-of-interest.cc/>

gitlab_connection *Connect to a specific gitlab instance API*

Description

Creates a function that can be used to issue requests to the specified gitlab API instance with the specified user private token and (for project_connection) only to a specified project.

Usage

```
gitlab_connection(gitlab_url, login = NULL, email = NULL, password = NULL,
  private_token = NULL, api_location = "/api/v3/")
```

```
project_connection(gitlab_url, project, login = NULL, email = NULL,
  password = NULL, private_token = NULL, api_location = "/api/v3/")
```

Arguments

gitlab_url	URL to the gitlab instance (e.g. https://gitlab.myserver.com)
login	name of user to login; either this or email or private token must be specified
email	email of user to login; either this or login or private token must be specified
password	password of user to login; if no private token but login or email is given, this must be specified
private_token	private_token with which to identify; either this or login/email + password must be specified to init connection
api_location	location of the gitlab API under the gitlab_url, usually and by default "/api/v3/"
project	id or name of project to issue requests to

Details

The returned function should serve as the primary way to access the gitlab API in the following. It can take vector/character arguments in the same way as the function `gitlab` does, as well as the convenience functions provided by this package or written by the user. If it is passed such that function it calls it with the arguments provided in `...` and the gitlab URL, api location and private_token provided when creating it via `gitlab_connection`.

Value

A function to access a specific gitlab API as a specific user, see details

Examples

```
## Not run:
my_gitlab <- gitlab_connection("http://gitlab.example.com", "123####89")
my_gitlab("projects")
my_gitlab(get_file, "test-project", "README.md", ref = "dev")

## End(Not run)
```

iff	<i>Apply a function if and only if test is TRUE</i>
-----	---

Description

otherwise return input value unchanged

Usage

```
iff(obj, test, fun, ...)
```

```
iffn(obj, test, fun, ...)
```

Arguments

obj	object to apply test and fun to
test	logical or function to apply to test
fun	function to apply
...	passed on to test

Details

iffn is ... if and only if test is FALSE

list_branches	<i>List, create and delete branches</i>
---------------	---

Description

List, create and delete branches

List, create and delete branches

List, create and delete branches

Usage

```
list_branches(project, verb = httr::GET, ...)
```

```
create_branch(project, branch_name, ref = "master", verb = httr::POST, ...)
```

```
delete_branch(project, branch_name, verb = httr::POST, ...)
```

Arguments

project	name or id of project (not repository!)
verb	is ignored, will always be forced to match the action the function name indicates
...	passed on to gitlab
branch_name	name of branch to create/delete
ref	ref name of origin for newly created branch

list_projects	<i>List projects in Gitlab</i>
---------------	--------------------------------

Description

List projects in Gitlab

Usage

```
list_projects(...)
```

Arguments

...	passed on to gitlab
-----	-------------------------------------

new_issue	<i>Post a new issue or edit one</i>
-----------	-------------------------------------

Description

Post a new issue or edit one

Post a new issue or edit one

Usage

```
new_issue(title, project, ...)
```

```
edit_issue(issue_id, project, ...)
```

```
close_issue(issue_id, project, ...)
```

```
reopen_issue(issue_id, project, ...)
```

```
assign_issue(issue_id, assignee_id = NULL, project, ...)
```

```
unassign_issue(issue_id, project, ...)
```

Arguments

title	title of the issue
project	project where the issue should be posted
...	further parameters passed to the API call, may contain description, assignee_id, milestone_id, labels, state_event (for edit_issue).
issue_id	id of issue to edit (projectwide iid, not gitlab API id)
assignee_id	numeric id of users as returned in '/users/' API request

pipe_into	<i>Pipe into specific formal argument</i>
-----------	---

Description

This rotates the order of the arguments such that the one named in param_name comes first and then calls the function.

Usage

```
pipe_into(x, param_name, fun, ...)
```

Arguments

x	value to be piped into fun
param_name	name of the argument that x should be assigned to
fun	function
...	further arguments for fun

prefix	<i>Prefix a string of text</i>
--------	--------------------------------

Description

Convenience function to use with magrittr wraps [paste0](#), hence vectorised as [paste0](#)

Usage

```
prefix(text, ...)
```

Arguments

text	goes to the end, rest
...	goes to the front.

proj_req	<i>Create a project specific request</i>
----------	--

Description

Prefixes the request location with "project/:id" and automatically translates project names into ids

Usage

```
proj_req(project, req, ...)
```

Arguments

project	project name or id
req	character vector of request location
...	passed on to get_project_id

push_file	<i>Upload a file to a gitlab repository</i>
-----------	---

Description

If the file already exists, it is updated/overwritten by default

Usage

```
push_file(project, file_path, content, commit_message, branch_name = "master",
  overwrite = TRUE, ...)
```

Arguments

project	Project name or id
file_path	path where to store file in repository
content	file content (text)
commit_message	Message to use for commit with new/updated file
branch_name	name of branch where to append newly generated commit with new/updated file
overwrite	whether to overwrite files that already exist
...	passed on to gitlab

Value

returns a data.frame with changed branch and path (0 rows if nothing was changed, since overwrite is FALSE)

repository	<i>Access to repository functions in Gitlab API</i>
------------	---

Description

Access to repository functions in Gitlab API

Usage

```
repository(req = c("tree"), project, ...)
```

```
list_files(...)
```

Arguments

req	request to perform on repository (everything after <code>/repository/</code> in gitlab API, as vector or part of URL)
project	name or id of project (not repository!)
...	passed on to gitlab API call, may include path argument for path

set_gitlab_connection	<i>Get/set a gitlab connection for all calls</i>
-----------------------	--

Description

This sets the default value of `gitlab_con` in a call to [gitlab](#)

Usage

```
set_gitlab_connection(gitlab_con = NULL, ...)
```

```
unset_gitlab_connection()
```

Arguments

gitlab_con	A function used for gitlab API calls, such as gitlab or as returned by gitlab_connection .
...	if <code>gitlab_con</code> is NULL, a new connection is created used the parameters is ... using gitlab_connection

to_issue_id	<i>Translate projectwide issue id to global gitlab API issue id</i>
-------------	---

Description

Translate projectwide issue id to global gitlab API issue id

Usage

```
to_issue_id(issue_id, project, ...)
```

Arguments

issue_id	projectwide issue id (as seen by e.g. gitlab website users)
project	project name or id
...	passed on to gitlab

Index

archive, 2
assign_issue (new_issue), 10

close_issue (new_issue), 10
comment_commit (get_comments), 4
comment_issue (get_comments), 4
create_branch (list_branches), 9
create_merge_request, 3

DELETE, 6
delete_branch (list_branches), 9

edit_comment (get_comments), 4
edit_commit_comment (get_comments), 4
edit_issue (new_issue), 10
edit_issue_comment (get_comments), 4

file_exists, 3

GET, 6
get_comments, 4
get_commit_comments (get_comments), 4
get_commits, 5
get_diff (get_commits), 5
get_file (file_exists), 3
get_issue (get_issues), 5
get_issue_comments (get_comments), 4
get_issues, 5
get_project_id, 6, 12
gitlab, 2–6, 6, 8, 10, 12–14
gitlab_connection, 6, 7, 13
gitlabr, 7
gitlabr-package (gitlabr), 7

iff, 9
iffn (iff), 9

list_branches, 9
list_files, 3
list_files (repository), 13
list_projects, 10

new_issue, 10

paste0, 11
pipe_into, 11
POST, 6
prefix, 11
proj_req, 12
project_connection (gitlab_connection),
7
push_file, 12
PUT, 6

reopen_issue (new_issue), 10
repository, 13

set_gitlab_connection, 13

to_issue_id, 14

unassign_issue (new_issue), 10
unset_gitlab_connection
(set_gitlab_connection), 13