Package: rmdgh (via r-universe)

December 31, 2024

Title R to GitHub productivity via Rmarkdown

Version 0.3.2

Description Browse and interact with GitHub via an Rmarkdown document interface.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Imports assertthat, fs, gert, reprex, rmarkdown (>= 2.15), xfun, yaml, backports, magrittr, atcursor (>= 0.0.2), clipr, curl, gh, glue, jsonlite, knitr, lubridate, prettyunits, rstudioapi, rvest, snakecase, tidyr, uuid, withr, utils, stats, usethis

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs git make libgit2-dev libicu-dev libxml2-dev libssl-dev libx11-dev

Repository https://milesmcbain.r-universe.dev

RemoteUrl https://github.com/milesmcbain/rmdgh

RemoteRef v0.3.2

RemoteSha 0c1553c4f18dd3ba594405da507b5daa5795944e

Contents

draft_issue *Create a GitHub issue from Rmarkdown*

Description

An Rmarkdown document is created an opened that can add or comment on issues/PRs when rendered with `rmarkdown::render()`, the 'knit' button in RStudio, or the 'Knit Rmd' command in VSCode.

Usage

```
draft_issue(
  filename = "issue.Rmd",
  path = getOption("rmdgh_issue_draft_path", get_pkg_user_dir()),
  overwrite = TRUE
)
```

Arguments

`filename`	the Rmarkdown file name to contain your issue
`path`	the path to create the rmd issue in. Defaults to `tools::R_user_dir("rmdgh")`. Issues created in the default directory will be automatically cleaned up. Default can be changed with option `rmd_gh_issue_draft_path`.
`overwrite`	whether or not to overwrite an existing issue with the same filename (defaults to TRUE).

Details

By default issues are created in a temporary dir, but this can be overidden and they will be created in the current directory.

get_gh_user *Get the github user using local git email config.*

Description

Get the github user using local git email config.

Usage

```
get_gh_user()
```

gh_thread *Open an issue thread*

Description

Three types of arugments are supported for opening an issue thread:

- no argument (NULL), an attempt is made to read a url for a thread from the clipboard
- numeric argument, an attempt is made to open an issue with this number in the repo corresponding to the current local repository
- A shorthand syntax: `"milesmcbain/capsule#12"` or `"capsule#12"`

Usage

```
gh_thread(thread = NULL)
```

Arguments

`thread`	text representing a thread to open, or nothing to try the clipboard. See details

github_issue *A github issue Rmarkdown format*

Description

A wraper for `rmarkdown::github_document()` that can peform actions on GitHub when rendering relating to issues and PRs including:

- create an issue
- update an issue title, body, and labels
- comment on an issue
- close an issue with a comment

Usage

```
github_issue(
  repo = NULL,
  number = NULL,
  labels = NULL,
  action = "create",
  draft = TRUE,
  close_with_comment = FALSE,
  fig_width = 7,
  fig_height = 5,
```

```
  dev = "png",
  df_print = "default",
  math_method = "default",
  wrap = "preserve"
)
```

Arguments

repo	the repository create the issue on e.g. "milesmcbain/capsule", or "capsule". Name will be resolved against locally insalled packages and CRAN in the later case.
number	the issue number in the repository if performing a comment or update action
labels	the labels to set for the issue if performing create or update action
action	the type of action to perform: "create", "update", "comment". Only "comment" is valid for PRs.
draft	if TRUE the action is not performed after rendering the document to GitHub markdown - set to TRUE initially to give you a chance to preview markdown output.
close_with_comment	if TRUE close the issue thread with comment action (assuming you have appropriate repo permissions).
fig_width	passed to image dimension passed to rmarkdown::github_document()
fig_height	image dimension passed to image dimension passed to rmarkdown::github_document()
dev	graphics device argument passed to rmarkdown::github_document()
df_print	data.frame print method passed to rmarkdown::github_document()
math_method	latex style math expression rendering method passed rmarkdown::github_document()
wrap	argument passed to pandoc. Controls use of line breaks. One of "auto", "preserve" or "none". "preserve" will mostly not interfere with your text formatting.

---

repo_issues *Search Issues and PRs and present results in Rmarkdown*

Description

issues() and its user-friendly wrappers are designed to allow you quick access to lists of issues and PRs that you can use as a jumping off point for exploring Rmarkdown issue threads.

Usage

```
repo_issues(
  repos = get_repo_remote(),
  query_description =
    glue::glue("repository issues for {paste(repos, collapse = \" \")}"),
  ...
```

```
)

repo_prs(
  repos = get_repo_remote(),
 query_description = glue::glue("repository PRs for {paste(repos, collapse = \" \")}"),
  ...
)

my_issues(
  author = get_gh_user(),
  query_description = glue::glue("{paste(author, collapse = \" \")} issues"),
  ...
)

issues_with_me(
  involves = get_gh_user(),
  query_description = glue::glue("issues with {paste0(involves)}"),
  ...
)

my_prs(
  author = get_gh_user(),
  type = "pr",
  query_description = glue::glue("PRs by {paste0(author)}"),
  ...
)

prs_with_me(
  involves = get_gh_user(),
  type = "pr",
  query_description = glue::glue("PRs with {paste0(involves)}"),
  ...
)

prs_for_me(
  user = get_gh_user(),
  type = "pr",
  query_description = glue::glue("PRs for {paste0(user)}"),
  extra_params = glue::glue("user:{user}"),
  ...
)

issues_for_me(
  user = get_gh_user(),
  type = "issue",
  query_description = glue::glue("Issues for {paste0(user)}"),
  extra_params = glue::glue("user:{user}"),
  ...
```

```
)

gh_for_me(
  user = get_gh_user(),
  type = NULL,
  query_description = glue::glue("Issues and PRs for {paste0(user)}"),
  extra_params = glue::glue("user:{user}"),
  ...
)

issues(
  repos = NULL,
  search_query = NULL,
  type = "issue",
  search_in = c("title", "body"),
  author = NULL,
  involves = NULL,
  is_open = TRUE,
  label = NULL,
  query_description = NULL,
  order = "desc",
  extra_params = NULL
)
```

Arguments

repos	a character vector issue repositories to get issues from. "owner/repo" and "repo" are allowed, with the repo resolved against installed R packages and CRAN in the latter case.
query_description	describe this query (appears in Rmarkdown results). Useful for building higher level functionality.
...	arguments passed to issues()
author	a character vector of issue authors to return issues by.
involves	a character vector of users to find in issue/PR threads, filtering returned results.
type	"issue" or "pr" or NULL for both kinds.
user	the user to return issues or PRs for
extra_params	a character vector of extra query parameters that are passed verbatim to the github API. This take the form "key:value" e.g. org:reditorsupport.
search_query	a character string to search for in issue titles and bodies.
search_in	where to apply search query. Any combination of "title" or "body". The default is c("title", "body") for both.
is_open	TRUE for open issues, FALSE for closed, NULL for both.
label	a character vector of issue labels to filter reutrned results.
order	"asc" or "desc" - the ordering of search results by last update date.

Details

Results are paged according to `getOption('issue_search_results_per_page')`.

Navigate between pages with `issue_search_results_forward()` and `issue_search_results_backward()`.

Previw an issue thread inline with search results with `issue_search_results_expand()`.

Use `jump_to_issue_thread()` to jump to an Rmarkdown thread based on the cursor position or `jump_to_issue_webpage()` to jump to the web.

Functions

- `repo_issues()`: issues for the local repository
- `repo_prs()`: PRs for the local repository
- `my_issues()`: issues authored by you
- `issues_with_me()`: issues referring to you
- `my_prs()`: PRs by you
- `prs_with_me()`: PRs referring to you
- `prs_for_me()`: PRs in repositories you own
- `issues_for_me()`: issues in repositories you own
- `gh_for_me()`: all issues and PRs in repositories you own.

`save_issue` *Save an issue to the current working directory*

Description

A helper for collecting issue documents locally to worked on at a later time.

Usage

```
save_issue(
  filename = NULL,
  folder = getOption("rmdgh_issue_location", "./issues")
)
```

Arguments

`filename`	the name to give the issue. Defaulted from issue repo and number in yaml.
`folder`	where to store the issue. The default is "./issues".

Details

`folder` will be `./issues` by default but is configurable in option `rmdgh_issue_location`. If it does not exist, it is created and added to the `.Rbuildignore`.

Index