GESTION DE GITLAB AUTOMATISÉE AVEC {GITLABR}

MEETUP R NANTES - 2022-11-21

SÉBASTIEN ROCHETTE - THINKR

Mise en place automatique d'un projet

- Développement et suivi de projet sur GitLab
- Un package interne avec {gitlabr} pour tout mettre en place

Démo - https://gitlab.com/statnmap

Ça veut dire quoi tout ça ?

Nous allons parler de :

- GitLab
- Gestion de projet
- {gitlabr}
- API GitLab

GitLab, une interface web pour votre projet git

- Un serveur git distant (sauvegarde)
- Un outil graphique de suivi des changements dans l'histoire
- Aide à la gestion de fusion de branches (Merge Request)

GitLab, un outil de gestion de projet

- Signalisation de bugs, demandes de fonctionnalités (*issues*)
- Communication entre développeurs.euses et utilisateurs.ices
- Suivi d'évolution de projet (Kanban)
- Documentation de projet (Wiki)
- Vérification automatique de code (Intégration Continue - CI)
- Publication automatique de pages HTML (Deploiement continu - CD)

{gitlabr} permet de gérer GitLab

- Commencer par mettre en place une connexion

```
# GitLab connexion
my_gitlab <- gl_connection(
  gitlab_url = "https://gitlab.com/",
  private_token = Sys.getenv("GITLAB_COM_TOKEN")
)

# Set the connection for the session
set_gitlab_connection(my_gitlab)
```

{gitlabr} permet de gérer GitLab

- Gestion de projet GitLab : `gl_*_project()`
- Gestion de ticket : `gl_*_issue()`
- Gestion de MR : `gl_*_issue()`

* pour `new`, `edit`, (`close`), `delete`

{gitlabr} gère le CI/CD

- Mise en place du fichier ".gitlab-ci.yml"
  - `use_gitlab_ci()`
  - `type = "check-coverage-pkgdown"`
  - `type = "check-coverage-pkgdown-renv"`
  - `type = "bookdown"`
  - `type = "bookdown-production"`
- Utilisation d'une branche de type "gh-pages"
  - https://github.com/statnmap/GitLab-Pages-Deploy
  - Une publication pour chaque branche importante

Utiliser la fonction `gitlab()`

- `gl_get_issues(project = 12)` appelle :

```
gitlab(
  req = c("projects", 12, "issues"),
  api_root = "https://gitlab.com/api/v4",
  private_token = "XXX", # authentication for API
  verb = httr::GET  # defaults to GET, but POST, PUT, DELETE can be used likewise
)
```

Vignette : https://statnmap.github.io/gitlabr/articles/d-go-further-understand-and-build.html

Utiliser les paramètres supplémentaires de l'API

- Utiliser les autres paramètres de l'API REST: https://docs.gitlab.com/ee/api/api_resources.html
- Exemple pour les issues : https://docs.gitlab.com/ee/api/issues.html

Attribute	Type	Required	Description
`assignee_id`	integer	no	Return issues assigned to the given user `id`. Mutually exclusive with `assignee_username`. `None` returns unassigned issues. `Any` returns issues with an assignee.
`assignee_username`	string array	no	Return issues assigned to the given `username`. Similar to `assignee_id` and mutually exclusive with `assignee_id`. In GitLab CE, the `assignee_username` array should only contain a single value. Otherwise, an invalid parameter error is returned.
`author_id`	integer	no	Return issues created by the given user `id`. Mutually exclusive with `author_username`. Combine with `scope=all` or `scope=assigned_to_me`.
`author_username`	string	no	Return issues created by the given `username`. Similar to `author_id` and mutually exclusive with `author_id`.

Utiliser les paramètres supplémentaires de l'API

- Avec une fonction {gitlabr} dédiée

```
gl_get_issues(
  project = 12,
  assignee_username = "statnmap",
  created_after = "2022-11-01T08:00:00Z", #ISO 8601
  state = "active"
)
```

- mais aussi avec `gitlab()`

```
gitlab(
  req = c("projects", 12, "issues"),
  assignee_username = "statnmap",
  created_after = "2022-11-01T08:00:00Z",
  verb = httr::GET,
  state = "active"
```

Répondre à vos besoins

- Créer vos propres fonctions avec `gitlab()`
- Par exemple, créer un wiki: https://docs.gitlab.com/ee/api/wikis.html#create-a-new-wiki-page

```
wiki_text <- c(
  "# Bienvenue sur le wiki",
  "",
  "Vous allez apprendre plein de choses ici")

# Home
wiki_home <- gitlab(
  req = paste0("projects/", project_id, "/wikis"),
  verb = httr::POST,
  content = paste(wiki_text, collapse = "\n"),
  title = "home",
  format = "markdown"
)
```

Aparté : {gitlabr} un package orphelin du CRAN

- {gitlabr}: Un package de Jirka Lewandowski
- Envoi d'un mail au CRAN pour éviter la sortie en "*orphan*"
- Envoi de votre version, le CRAN envoie un mail aux "*maintainers*" originaux pour validation
- Si le package est déjà "*orphan*", il suffit d'envoyer une nouvelle version

N'hésitez pas, si votre package préféré est abandonné, vous pouvez le reprendre

A vous de jouer !

- Mise en place de projet avec {gitlabr} : https://statnmap.github.io/gitlabr/articles/b-projects.html
- Des fonctions prêtes à documenter, tester
  - “*project issue state events*”, “*wiki*”, “*protect branches*”,
  - https://github.com/statnmap/gitlabr/blob/main/dev/upcomi
- Plein d’issues et PR ouverts : https://github.com/statnmap/gitlabr/issues