Package: ghapps (via r-universe)

March 2, 2025

Type Package

Title Authenticate as a 'GitHub' App

Version 1.1.1

Description 'GitHub' apps provide a powerful way to manage fine grained programmatic access to specific 'git' repositories, without having to create dummy users, and which are safer than a personal access token for automated tasks. This package extends the 'gh' package to let you authenticate and interact with 'GitHub' <https://docs.github.com/en/rest/overview> in 'R' as an app.

License MIT + file LICENSE

URL https://r-lib.r-universe.dev/ghapps

BugReports https://github.com/r-lib/ghapps/issues

Encoding UTF-8

RoxygenNote 7.3.2.9000

Imports gh, jose, openssl

Roxygen list(markdown = TRUE)

Config/pak/sysreqs libssl-dev

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

RemoteUrl https://github.com/r-lib/ghapps

RemoteRef HEAD

RemoteSha af331fe9917ddf29ee2187cf1f4b04d3b080eb56

Contents

ghapps *Authenticate as a GitHub App*

Description

GitHub *apps* provide a powerful way to manage fine grained programmatic access to specific git repositories, without having to create dummy users, and which are safer than PATs for automated tasks. This package extends gh::gh to let you authenticate and interact with the GitHub API in R on behalf of an app.

Usage

```
gh_app_info(jwt = gh_app_jwt())

gh_app_token(installation, jwt = gh_app_jwt())

gh_app_installation_count(jwt = gh_app_jwt())

gh_app_installation_list(jwt = gh_app_jwt())

gh_app_installation_info(installation, jwt = gh_app_jwt())

gh_app_installation_repositories(installation, jwt = gh_app_jwt())

gh_app_installation_delete(installation, jwt = gh_app_jwt())

gh_app_public_info(name = "r-universe")

gh_app_jwt(
  app_id = Sys.getenv("GH_APP_ID"),
  app_key = Sys.getenv("GH_APP_KEY")
)

gh_demo_jwt()
```

Arguments

jwt	a jwt credential object generated with gh_app_jwt()
installation	the target repositorie(s) which we want to access. Either a user / organization name such as "ropensci", or a specific repository that has the app installed for example "ropensci/magick".
name	slug of the github app from https://github.com/apps/{name}
app_id	a string with the github app id
app_key	file path or string with your private key, passed to openssl::read_key. or a key object returned by openssl::read_key

Details

Instead of authenticating as a user, you can also authenticate with the GitHub API as a "GitHub app". An app is a first class actor within GitHub. This means it has its own permissions to specific repositories, without being tied to any particular user account. In fact, you could think of an app as a special type of dummy user with a specific purpose. Don't be too intimidated by the word 'app', it is mainly an authentication concept. Any program that authenticates with GitHub, such as an R script, can be considered an app.

Creating a GitHub app:

To register a new app go to: https://github.com/settings/apps/new. You may register as many apps as you like, with different permissions, for different purposes.

You can choose if the the app is public (to allow others to give the app access to their repo by "installing" the app) or if the app is only for your own account. The latter is basically a safe way to provide limited access to a single repository, without exposing your user account in any way.

Acting on behalf of an app in R:

We can interact with most of the GitHub API on behalf of an app in exactly the same way as a regular user. The only difference is that instead of a personal access token (PAT) we generate a temporary app token, which looks very similar. From here we use `gh::gh()` in exactly the same way as usual.

Authentication on behalf of an app is a two step process. First you need to generate a so-called JWT with `gh_app_jwt()` using the App-ID and a RSA key file that you can retrieve on GitHub in the app settings. This JWT is only valid for 5 minutes, and is used to generate app tokens for specific target repositories with `gh_app_token()`. This app token works the same as a PAT, but it is valid for 1 hour and has permission only to the repository that you specified as the `installations` parameter in `gh_app_token()`. From here you can use `gh::gh()` to perform all the operations that the git repository has given your app permission for.

Using this in CI:

A common use case is to authenticate as a github-app inside a github action script, in order to run tasks that require authentication, without the need to use someones personal credentials. To make this easy, you can specify the app-id and private key via environment variables `GH_APP_ID` and `GH_APP_KEY` which you can expose as a 'secret' in the CI. The `GH_APP_KEY` can either be a file path, or simply the verbatim content of the private key (pem) file.

Value

`gh_app_token` returns a temporary token that will be valid for 1 hour that you use instead of a PAT.

Examples

```
# Authenitcate and show some metadata about our demo-app
gh_app_info(jwt = gh_demo_jwt())
gh_app_installation_list(jwt = gh_demo_jwt())

# This requires that user 'testingjerry' has our app installed:
gh_app_installation_info('testingjerry', jwt = gh_demo_jwt())
token <- gh_app_token('testingjerry', jwt = gh_demo_jwt())
```

```
# Use the token in gh() to do things on behalf of the app
res <- gh::gh('/users/testingjerry', .token = token)
```

Index