

# Package: GithubActions (via r-universe)

May 26, 2026

**Type** Package

**Title** Functions to facilitate use of Github Actions via R

**Version** 0.1.0

**Author** Sebastian Kranz

**Maintainer** Sebastian Kranz <sebastian.kranz@uni-ulm.de>

**Description** Work in progress. Not yet working well.

**License** GPL >= 2.0

**Depends** httr, curl, jsonlite, restorepoint

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Config/pak/sysreqs** libssl-dev

**Repository** <https://skranz.r-universe.dev>

**Date/Publication** 2026-03-27 06:15:13 UTC

**RemoteUrl** <https://github.com/repboxr/GithubActions>

**RemoteRef** main

**RemoteSha** 95099ea93b021e60b33246589308c74b3e653cc3

## Contents

gh_auth_status . . . . .	2
gh_clone . . . . .	2
gh_download_artifact . . . . .	3
gh_info_run . . . . .	3
gh_list_artifacts . . . . .	3
gh_list_run_jobs . . . . .	4
gh_list_runs . . . . .	4
gh_list_workflows . . . . .	4
gh_login . . . . .	4
gh_new_repo . . . . .	5

gh_newest_runid . . . . .	5
gh_pat . . . . .	5
gh_remove_branch . . . . .	6
gh_remove_history . . . . .	6
gh_remove_secret . . . . .	6
gh_run_log . . . . .	6
gh_run_workflow . . . . .	7
gh_set_pat . . . . .	7
gh_set_secret . . . . .	8
gh_update . . . . .	8
gh_wait_until_new_run_starts . . . . .	8
gh_wait_until_run_completed . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

gh_auth_status	<i>Show authentication status</i>
----------------	-----------------------------------

---

### Description

Show authentication status

### Usage

gh\_auth\_status()

---

gh_clone	<i>Clone a repository on Github to a local directory</i>
----------	--

---

### Description

Clone a repository on Github to a local directory

### Usage

gh\_clone(repo, repodir)

---

gh\_download\_artifact     *Download an artifact generated by a finished job*

---

**Description**

Download an artifact generated by a finished job

**Usage**

```
gh_download_artifact(  
  repo,  
  destfile,  
  artifact_id = NULL,  
  runid = NULL,  
  pat = gh_pat()  
)
```

---

gh\_info\_run             *Info about a workflow run.*

---

**Description**

Info about a workflow run.

**Usage**

```
gh_info_run(repo, runid, pat = gh_pat())
```

---

gh\_list\_artifacts       *List all artifacts of a repo List all artifacts of a repo or single run*

---

**Description**

List all artifacts of a repo List all artifacts of a repo or single run

**Usage**

```
gh_list_artifacts(repo, runid = NULL, pat = gh_pat())
```

---

gh_list_run_jobs	<i>Show all jobs of a run. Some functions require the corresponding job id.</i>
------------------	---

---

**Description**

Show all jobs of a run. Some functions require the corresponding job id.

**Usage**

```
gh_list_run_jobs(repo, runid, pat = gh_pat())
```

---

gh_list_runs	<i>Return all workflow runs</i>
--------------	---------------------------------

---

**Description**

Return all workflow runs

**Usage**

```
gh_list_runs(repo, pat = gh_pat())
```

---

gh_list_workflows	<i>Return all workflows of the specified repo</i>
-------------------	---

---

**Description**

Return all workflows of the specified repo

**Usage**

```
gh_list_workflows(repo, pat = gh_pat())
```

---

gh_login	<i>Login to github with your email and username</i>
----------	---

---

**Description**

Login to github with your email and username

**Usage**

```
gh_login(email, username)
```

---

gh_new_repo	<i>Create a new repo on github and locally</i>
-------------	--

---

**Description**

Create a new repo on github and locally

**Usage**

```
gh_new_repo(  
  reponame,  
  parentdir,  
  access = c("public", "private")[2],  
  pat = gh_pat()  
)
```

---

gh_newest_runid	<i>Get the id if the newest workflow run of a repo.</i>
-----------------	---

---

**Description**

Get the id if the newest workflow run of a repo.

**Usage**

```
gh_newest_runid(repo, pat = gh_pat())
```

---

gh_pat	<i>Get personal access token that was previously set b gh_set_pat.</i>
--------	--

---

**Description**

Get personal access token that was previously set b gh\_set\_pat.

**Usage**

```
gh_pat()
```

---

gh\_remove\_branch      *Remove a branch from local and github repository*

---

**Description**

Remove a branch from local and github repository

**Usage**

```
gh_remove_branch(repodir, branch)
```

---

gh\_remove\_history      *Removes all old commits of a branch and only keeps current local files*

---

**Description**

Based on <https://superuser.com/questions/926915/how-can-i-delete-old-commits-in-github-via-terminal>  
WARNING: Don't use for real code repos, in particular not for repos multiple users work with.

**Usage**

```
gh_remove_history(repodir, branch = "main")
```

---

gh\_remove\_secret      *Remove a secret from the repository*

---

**Description**

Remove a secret from the repository

**Usage**

```
gh_remove_secret(repodir, name)
```

---

gh\_run\_log      *Get log of a particular workflow run*

---

**Description**

Get log of a particular workflow run

**Usage**

```
gh_run_log(repodir, runid)
```

---

gh_run_workflow	<i>Trigger the specified workflow</i>
-----------------	---------------------------------------

---

### Description

Only works if the workflow YAML file sets

### Usage

```
gh_run_workflow(  
  repo,  
  name = NULL,  
  branch = "main",  
  inputs = NULL,  
  pat = gh_pat()  
)
```

### Arguments

repo	The repository name in format user/reponame.
name	The name of the workflow. If NULL use the first workflow
branch	The Github branch. By default "main"
inputs	Not yet tested. Potential input parameters passed to the workflow.

### Details

on: workflow\_dispatch:

---

gh_set_pat	<i>Set personal access token by writing it into environment variables.</i>
------------	--

---

### Description

Set personal access token by writing it into environment variables.

### Usage

```
gh_set_pat(pat)
```

### Arguments

pat	A string containing the personal access token generated on Github. Not very secure. At some point I should use the approach of the gh package
-----	--

---

gh_set_secret	<i>Set a repository secret using github client</i>
---------------	--

---

**Description**

Set a repository secret using github client

**Usage**

```
gh_set_secret(repodir, name, value)
```

**Arguments**

repodir	path to your local repository
name	name of the secret
value	of the secret

---

gh_update	<i>Updates the Github repository to the new content of the local repo directory</i>
-----------	---

---

**Description**

Updates the Github repository to the new content of the local repo directory

**Usage**

```
gh_update(repodir, msg = "update", remote = "origin", branch = "main")
```

---

gh_wait_until_new_run_starts	<i>Wait until a new Github action job has started</i>
------------------------------	---

---

**Description**

Wait until a new Github action job has started

**Usage**

```
gh_wait_until_new_run_starts(
  repo,
  old_runid,
  pause.sec = 1,
  timeout = 60,
  pat = gh_pat()
)
```

---

`gh_wait_until_run_completed`

*Wait until a new Github action job is completed*

---

### **Description**

Wait until a new Github action job is completed

### **Usage**

```
gh_wait_until_run_completed(  
  repo,  
  runid,  
  pause.sec = 1,  
  timeout = 60 * 60 * 10,  
  pat = gh_pat()  
)
```

# Index

gh\_auth\_status, 2  
gh\_clone, 2  
gh\_download\_artifact, 3  
gh\_info\_run, 3  
gh\_list\_artifacts, 3  
gh\_list\_run\_jobs, 4  
gh\_list\_runs, 4  
gh\_list\_workflows, 4  
gh\_login, 4  
gh\_new\_repo, 5  
gh\_newest\_runid, 5  
gh\_pat, 5  
gh\_remove\_branch, 6  
gh\_remove\_history, 6  
gh\_remove\_secret, 6  
gh\_run\_log, 6  
gh\_run\_workflow, 7  
gh\_set\_pat, 7  
gh\_set\_secret, 8  
gh\_update, 8  
gh\_wait\_until\_new\_run\_starts, 8  
gh\_wait\_until\_run\_completed, 9