

# Package: ghe.utils (via r-universe)

October 15, 2024

**Title** Utilities for Working with GitHub Enterprise  
**Version** 0.1.2  
**Description** Set the location of your company's GitHub Enterprise address, and seamlessly interact with a GitHub Enterprise Instance from the R console.  
**License** MIT + file LICENSE  
**Depends** R (>= 3.4)  
**Imports** curl (>= 4.3.2), cli (>= 3.1.0), usethis (>= 2.1.3), gh (>= 1.3.0), remotes (>= 2.4.1)  
**Encoding** UTF-8  
**Roxygen** list(markdown = TRUE)  
**RoxygenNote** 7.1.2  
**URL** <http://www.danieldsjoberg.com/ghe.utils>  
**Suggests** spelling  
**Language** en-US  
**Repository** <https://ddsjoberg.r-universe.dev>  
**RemoteUrl** <https://github.com/ddsjoberg/ghe.utils>  
**RemoteRef** v0.1.2  
**RemoteSha** b17555ebfcadf406d19bc1e37be78aa7e48d8923

## Contents

ghe_host . . . . .	2
has_ghe_network_access . . . . .	2
install_ghe . . . . .	3
token . . . . .	4
use_ghe . . . . .	4
use_ghe_date_release . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

ghe_host	<i>Retrieve GH Enterprise Location</i>
----------	--

---

**Description**

Retrieve GH Enterprise Location

**Usage**

ghe\_host()

---

has_ghe_network_access	<i>Check for GHE Network Access</i>
------------------------	-------------------------------------

---

**Description**

Check for GHE Network Access

**Usage**

has\_ghe\_network\_access()

**Value**

logical

**Author(s)**

Daniel D. Sjoberg

**Examples**

has\_ghe\_network\_access()

install\_ghe

*Install a package directly from GitHub Enterprise.***Description**

Install a package directly from GitHub Enterprise.

**Usage**

```
install_ghe(repo, ...)
```

**Arguments**

repo	Repository address in the format <code>username/repo[/subdir][@ref #pull @*release]</code> . Alternatively, you can specify <code>subdir</code> and/or <code>ref</code> using the respective parameters (see below); if both are specified, the values in <code>repo</code> take precedence.
...	Arguments passed on to <code>remotes::install_github</code>
ref	Desired git reference. Could be a commit, tag, or branch name, or a call to <code>github_pull()</code> or <code>github_release()</code> . Defaults to "HEAD", which means the default branch on GitHub and for git remotes. See <a href="#">setting-the-default-branch</a> for more details.
subdir	Subdirectory within <code>repo</code> that contains the R package.
dependencies	Which dependencies do you want to check? Can be a character vector (selecting from "Depends", "Imports", "LinkingTo", "Suggests", or "Enhances"), or a logical vector. TRUE is shorthand for "Depends", "Imports", "LinkingTo" and "Suggests". NA is shorthand for "Depends", "Imports" and "LinkingTo" and is the default. FALSE is shorthand for no dependencies (i.e. just check this package, not its dependencies). The value "soft" means the same as TRUE, "hard" means the same as NA. You can also specify dependencies from one or more additional fields, common ones include: <ul style="list-style-type: none"> <li>• <code>Config/Needs/website</code> - for dependencies used in building the pkgdown site.</li> <li>• <code>Config/Needs/coverage</code> for dependencies used in calculating test coverage.</li> </ul>
upgrade	Should package dependencies be upgraded? One of "default", "ask", "always", or "never". "default" respects the value of the <code>R_REMOTES_UPGRADE</code> environment variable if set, and falls back to "ask" if unset. "ask" prompts the user for which out of date packages to upgrade. For non-interactive sessions "ask" is equivalent to "always". TRUE and FALSE are also accepted and correspond to "always" and "never" respectively.
force	Force installation, even if the remote state has not changed since the previous install.
quiet	If TRUE, suppress output.

build If TRUE build the package before installing.  
 build\_opts Options to pass to R CMD build, only used when build is TRUE.  
 build\_manual If FALSE, don't build PDF manual ('-no-manual').  
 build\_vignettes If FALSE, don't build package vignettes ('-no-build-vignettes').  
 repos A character vector giving repositories to use.  
 type Type of package to update.

**Author(s)**

Daniel D. Sjoberg

**See Also**

[remotes::install\\_github\(\)](#)

---

token	<i>Retrieve GH Enterprise Personal Access Token (PAT)</i>
-------	---

---

**Description**

Functions for setting up a GHE PAT (`create_ghe_token()`), and accessing the saved PAT (`ghe_token()`).

**Usage**

```

ghe_token()

create_ghe_token(scopes = c("repo", "user", "gist", "workflow"))

```

**Arguments**

scopes Character vector of token scopes, pre-selected in the web form. Final choices are made in the GitHub form. Read more about GitHub API scopes at <https://docs.github.com/apps/building-oauth-apps/understanding-scopes-for-oauth-apps/>.

---

use_ghe	<i>Connect a local repo with GitHub Enterprise</i>
---------	--

---

**Description**

Function in a wrapper for `usethis::use_github()` with the GitHub Enterprise instance.

**Usage**

```
use_ghe(...)
```

**Arguments**

- ... Arguments passed on to `usethis::use_github`
- `organisation` If supplied, the repo will be created under this organisation, instead of the login associated with the GitHub token discovered for this host. The user's role and the token's scopes must be such that you have permission to create repositories in this organisation.
- `private` If TRUE, creates a private repository.
- `visibility` Only relevant for organisation-owned repos associated with certain GitHub Enterprise products. The special "internal" visibility grants read permission to all organisation members, i.e. it's intermediate between "private" and "public", within GHE. When specified, `visibility` takes precedence over `private = TRUE/FALSE`.
- `protocol` One of "https" or "ssh"

**See Also**

Review `usethis::use_github()` for details.

---

use\_ghe\_date\_release *Make GHE Release*

---

**Description**

Make GHE Release

**Usage**

```
use_ghe_date_release(name = "<release annotation>", append_date = TRUE)
```

**Arguments**

- `name` Name of GHE release
- `append_date` Logical indicating whether to append the date to the release name.

# Index

`create_ghe_token(token)`, 4

`ghe_host`, 2

`ghe_token(token)`, 4

`github_pull()`, 3

`github_release()`, 3

`has_ghe_network_access`, 2

`install_ghe`, 3

`remotes::install_github`, 3

`remotes::install_github()`, 4

`token`, 4

`use_ghe`, 4

`use_ghe_date_release`, 5

`usethis::use_github`, 5

`usethis::use_github()`, 5