

Package ‘natmanager’

July 6, 2020

Title Install the 'Natverse' Packages from Scratch

Version 0.4.6

URL <https://github.com/natverse/natmanager>

BugReports <https://github.com/natverse/natmanager/issues>

Description Provides streamlined installation for packages from the 'natverse', a suite of R packages for computational neuroanatomy built on top of the 'nat' 'NeuroAnatomy Toolbox' package. Installation of the complete 'natverse' suite requires a 'GitHub' user account and personal access token 'GITHUB_PAT'. 'natmanager' will help the end user set this up if necessary.

License GPL-3

Imports gh, utils, remotes, usethis, curl, withr

Suggests testthat (>= 2.1.0), knitr, rstudioapi

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Sridhar Jagannathan [aut, cre]
(<<https://orcid.org/0000-0002-2078-1145>>),
Gregory Jefferis [aut] (<<https://orcid.org/0000-0002-0587-9355>>)

Maintainer Sridhar Jagannathan <j.sridharrajan@gmail.com>

Repository CRAN

Date/Publication 2020-07-06 17:20:02 UTC

R topics documented:

install	2
list_repo	3
selfupdate	4

Index	5
--------------	----------

install	<i>Install natverse packages from GitHub</i>
---------	--

Description

install allows you to install one of two collections of nat packages

- core a minimal install that can help users to get started with nat and already solve many problems (the default)
- natverse a powerful "batteries included" distribution with all mature packages in the natverse.

Since the natverse option will install many packages from GitHub, you need to have a GitHub account and personal access token (GITHUB_PAT). Install will check to see if you have a GITHUB_PAT already and, if not, walk you through the steps of setting one up. A fall-back PAT is built into the package but we strongly recommend that you sign up to GitHub and get your own if you start using the natverse regularly.

check_pat can be used to check if you have a GITHUB_PAT set and will advise on how to do this if necessary.

Usage

```
install(
  collection = c("core", "natverse"),
  pkgs = NULL,
  dependencies = TRUE,
  upgrade.dependencies = "always",
  ...
)

check_pat(create = TRUE)
```

Arguments

collection	The collection of natverse packages that you would like to install. The current options are core, the default, or natverse. See Description for more information.
pkgs	A character vector of package names specifying natverse packages to install. When present overrides the collection argument.
dependencies	Which dependencies you want to install see install_github .
upgrade.dependencies	Whether to install dependencies of natmanager. See the upgrade argument of install_github for details. The default will go ahead and always do this is as necessary.
...	extra arguments to pass to install_github .

create Whether to help you create a personal GITHUB_PAT if you do not have one set. When create=FALSE a default PAT will be used if you have not set your own. This could cause trouble if other people are using the same PAT.

Value

check_pat returns the PAT invisibly or errors out if create=TRUE and none can be set.

Examples

```
## Not run:
# install core packages to try out the core natverse
natmanager::install('core')

# Full "batteries included" installation with all packages
natmanager::install('natverse')

# Install natverse, non-natverse package
# for natverse packages no need to specify the repo
natmanager::install(pkgs=c('nat.jrcbrains', 'flyconnectome/hemibrainr'))

## End(Not run)
## Not run:
# Check status of GitHub PAT and create one if required
natmanager::check_pat(create=TRUE)
# Check status of GitHub PAT and use default if no personal one available
natmanager::check_pat(create=FALSE)

## End(Not run)
```

list_repo	<i>List all the repos inside a particular GitHub organisation</i>
-----------	---

Description

by default this will list all the repositories inside the ‘natverse’ organization.

Usage

```
list_repo(orgname = "natverse")
```

Arguments

orgname Name of the GitHub organization

Value

Character vector of repository names

Examples

```
natmanager::list_repo()
```

selfupdate	<i>Update the 'natmanager' package itself.</i>
------------	--

Description

Update the 'natmanager' package itself.

Usage

```
selfupdate(
  source = c("GITHUB", "CRAN"),
  upgrade.dependencies = "always",
  force = FALSE,
  ...
)
```

Arguments

source	Location from which to obtain a newer version of natmanager. Defaults to GITHUB since this may well have a newer version than the CRAN package repository.
upgrade.dependencies	Whether to install dependencies of natmanager. See the upgrade argument of install_github for details. The default will go ahead and always do this is as necessary.
force	Force self update even if there doesn't seem to be an update (default FALSE)
...	extra arguments to pass to install_github or install_cran .

Value

Logical indicating whether an update was required (invisibly).

See Also

[install](#)

Examples

```
## Not run:
natmanager::selfupdate()

## End(Not run)
```

Index

`check_pat (install)`, [2](#)

`install`, [2](#), [4](#)

`install_cran`, [4](#)

`install_github`, [2](#), [4](#)

`list_repo`, [3](#)

`selfupdate`, [4](#)