Package ‘batata’

October 12, 2022

Type Package

Title Managing Packages Removal and Installation

Version 0.2.1

Author Mohamed El Fodil Ihaddaden

Maintainer Mohamed El Fodil Ihaddaden <ihaddaden.fodeil@gmail.com>

Description Allows the user to manage easily R packages removal and installation. It offers many functions to display installed packages according to
specific dates and removes them if needed. The user is always prompted when running the removal functions in order to confirm
the required action. It also provides functions that will install 'Github' starred R packages whether available on 'CRAN' or not.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports fs, utils, glue, lubridate, jsonlite, remotes, purrr

RoxygenNote 7.1.1

URL https://github.com/feddelegrand7/batata

BugReports https://github.com/feddelegrand7/batata/issues

Suggests knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-03-08 09:50:02 UTC

R topics documented:

`display_most_starred` *Display the most starred R Github Repositories*

Description

Display the most starred R Github Repositories

Usage

```
display_most_starred(n = 10)
```

Arguments

`n`	the number of most starred Github R repositories to fetch. Defaults to 10.

Value

a character vector of the most starred R repositories

`display_starred` *Display User's Github Starred Repositories*

Description

Display User's Github Starred Repositories

Usage

```
display_starred(github_user, n = 5, onlyR = FALSE)
```

Arguments

`github_user` the Github user name to look for

`n` the number of the last starred repositories. Defaults to 5 in which case it will return the last 5 starred repositories. Note that if the 'onlyR' parameter is set to TRUE, you might get a lower number of starred repos due to filtering R from all the other languages.

`onlyR` Logical, whether to fetch only R repositories, Default to FALSE

Value

A character vector of starred Github repositories

`fresh_start` *Remove all the installed R packages from a specified library*

Description

Remove all the installed R packages from a specified library

Usage

```
fresh_start(lib = .libPaths())
```

Arguments

`lib` a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

called for the side effect of removing all installed packages

Examples

```

## Not run:

# DANGER: THE FUNCTION REMOVES ALL THE PACKAGES

fresh_start()


## End(Not run)
```

`install_most_starred` *Install the most starred CRAN packages*

Description

Install the most starred CRAN packages

Usage

```
install_most_starred(n = 10)
```

Arguments

`n`	the most starred starred CRAN packages. Defaults to 10. in this case the function will look at the 10 most starred R repo and install them if available on CRAN.

Value

called for the side effect of installing most starred CRAN packages

`install_starred_cran` *Install Github Starred CRAN Packages*

Description

installs the Github starred packages from CRAN

Usage

```
install_starred_cran(github_user, n = 5)
```

Arguments

`github_user`	the Github user name to look for
`n`	the last 'n' starred repositories. Defaults to 5, in which case it will look for the last 5 starred repositories, filter the R repos and install them

Value

called for the side effect of installing the Github starred packages that are available on CRAN

`install_starred_github` *Install Github Starred Packages from Github*

Description

installs the Github starred repositories from Github and not from CRAN.

Usage

```
install_starred_github(github_user, n = 5, upgrade = "never")
```

Arguments

`github_user` the Github user name to look for

`n` the last 'n' starred repositories. Defaults to 5, in which case it will look for the last 5 starred repositories, filter the R repos and install them

`upgrade` whether to upgrade out of date packages. You can choose from 'always' or 'never'. Defaults to 'never'. For more info, see <install_github()> from the 'remote' package.

Value

called for the side effect of installing the Github starred repositories

`latest_packages` *Displaying the latest installed R packages*

Description

Displaying the latest installed R packages

Usage

```
latest_packages(n = 1, lib = .libPaths())
```

Arguments

`n` the number of the last installed packages to display. Default to n = 1, will return the last installed package

`lib` a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

a data frame

Examples

```
## Not run:

# Displaying the last 10 installed packages
latest_packages(10)


## End(Not run)
```

rm_latest_packages *Remove the n latest installed R packages*

Description

Remove the n latest installed R packages

Usage

```
rm_latest_packages(n = 1, lib = .libPaths())
```

Arguments

`n`	the last number of installed packages to remove. Default to 1 for the last installed package
`lib`	a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

called for the side effect of removing the n latest installed packages

Examples

```

## Not run:

# Removing the last 10 installed packages

rm_latest_packages(n = 10)


## End(Not run)
```

rm_since_packages *removes installed packages according to a specific date*

Description

removes installed packages according to a specific date

Usage

```
rm_since_packages(date, position, lib = .libPaths())
```

Arguments

`date`	the date of interest in yyyy-mm-dd format
`position`	takes three arguments "at", "before" or "after". "at" displays the packages installed at the chosen date, "before" before that date and "after" after that date)
`lib`	a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

a character vector

Examples

```
## Not run:
# Displaying the packages installed today
today_packages()

## End(Not run)
```

rm_today_packages *Remove the packages installed in the current day*

Description

Remove the packages installed in the current day

Usage

```
rm_today_packages(lib = .libPaths())
```

Arguments

`lib`	a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

called for the side effect of removing the today installed packages

Examples

```
## Not run:

# Removing the packages installed today

rm_today_packages()

## End(Not run)
```

rm_yesterday_packages *Remove the packages installed yesterday*

Description

Remove the packages installed yesterday

Usage

```
rm_yesterday_packages(lib = .libPaths())
```

Arguments

`lib`	a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

called for the side effect of removing the yesterday installed packages

Examples

```
## Not run:
# Removing the packages installed yesterday
rm_yesterday_packages()

## End(Not run)
```

since_packages *Displays installed packages according to a specific date*

Description

Displays installed packages according to a specific date

Usage

```
since_packages(date, position, lib = .libPaths())
```

Arguments

date	the date of interest in yyyy-mm-dd format
position	takes three arguments "at", "before" or "after". "at" displays the packages installed at the chosen date, "before" before that date and "after" after that date)
lib	a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

a character vector

Examples

```
## Not run:
# Displaying the packages installed today
since_packages(date = Sys.Date(), position = "at")

## End(Not run)
```

today_packages *Displays the packages installed in the current day*

Description

Displays the packages installed in the current day

Usage

```
today_packages(lib = .libPaths())
```

Arguments

lib	a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

a character vector

Examples

```
## Not run:
# Displaying the packages installed today
today_packages()

## End(Not run)
```

`yesterday_packages` *Displays the packages installed yesterday*

Description

Displays the packages installed yesterday

Usage

```
yesterday_packages(lib = .libPaths())
```

Arguments

`lib`	a character vector giving the library directories. Defaults to the first element in .libPaths()

Value

a character vector

Examples

```
## Not run:
# Displaying the packages installed yesterday
yesterday_packages()

## End(Not run)
```

Index