Package 'starter'

October 15, 2024

Title Starter Kit for New Projects

Version 0.1.16

Description Get started with new projects by dropping a skeleton of a new project into a new or existing directory, initialise git repositories, and create reproducible environments with the 'renv' package. The package allows for dynamically named files, folders, file content, as well as the functionality to drop individual template files into existing projects.

License AGPL (>= 3)

URL https://github.com/ddsjoberg/starter, https://www.danieldsjoberg.com/starter/index.html

BugReports https://github.com/ddsjoberg/starter/issues

Depends R (>= 3.6)

Imports dplyr (>= 1.1.0), cli (>= 3.6.0), gert (>= 1.9.2), glue (>= 1.6.2), R.utils (>= 2.11.0), renv (>= 0.17.2), rlang (>= 1.0.6), rprojroot, rstudioapi

Suggests covr (>= 3.5.1), fs (>= 1.5.0), knitr (>= 1.34), pkgdown (>= 1.6.1), readr (>= 2.0.1), rmarkdown (>= 2.11), stringr, spelling (>= 2.2), testthat (>= 3.0.4)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Daniel D. Sjoberg [aut, cre] (<https://orcid.org/0000-0003-0862-2018>), Emily Vertosick [ctb]

Maintainer Daniel D. Sjoberg <danield.sjoberg@gmail.com>

Repository CRAN

Date/Publication 2024-10-15 16:20:01 UTC

Contents

`create_project` *Start a new project*

Description

Creates a directory with the essential files for a new project. The function can be used on existing project directories as well. Existing files will *not* be overwritten; rather, the user will be prompted whether to replace the existing file with the template file.

Usage

```
create_project(
  path,
  path_data = NULL,
  template = "default",
  git = TRUE,
  renv = TRUE,
  symlink = git,
  renv.settings = NULL,
  overwrite = NA,
  open = interactive()
)
```

Arguments

`path`	A path. If it exists, it is used. If it does not exist, it is created.
`path_data`	A path. The directory where the secure data exist. Default is `NULL`. When supplied, a symbolic link to data folder will be created.
`template`	A project template. See vignette for details.
`git`	Logical indicating whether to create Git repository. Default is `TRUE` When `NA`, user will be prompted whether to initialise Git repo.
`renv`	Logical indicating whether to add renv to a project. Default is `TRUE`. When `NA` user is asked interactively for preference.
`symlink`	Logical indicating whether to place a symbolic link to the location in `path_data=`. Default is to place the symbolic link if the project is a git repository.
`renv.settings`	A list of renv settings passed to `renv::scaffold(settings=)`
`overwrite`	Logical indicating whether to overwrite existing files if they exist. Options are `TRUE`, `FALSE`, and `NA` (aka ask interactively). Default is `NA`
`open`	Logical indicating whether to open new project in fresh RStudio session

Value

NULL, places project template in new or existing directory

Personalized Template

Users can create a personalized project template. Check out the vignette for step by step instructions.

Author(s)

Daniel D. Sjoberg

See Also

`use_project_file()`

Vignette for `create_project()`

Examples

```
# specifying project folder location (folder does not yet exist)
project_path <- file.path(tempdir(), "My Project Folder")

# creating folder where secure data would be stored (typically will be a network drive)
secure_data_path <- file.path(tempdir(), "secure_data")
dir.create(secure_data_path)

# creating new project folder
create_project(project_path, path_data = secure_data_path)
```

`create_symlink` *Establish symbolic link between folders*

Description

The `starter_symlink()` function is an OS agnostic function that creates symbolic links between two folders. The function is, at its core, a wrapper for the `R.utils::createLink()` function with opinionated defaults. The function must be called in an environment where the working directory is known (e.g. using `*.Rproj`, `setwd()`, etc.).

Usage

```
create_symlink(to, name = "secure_data", ...)
```

Arguments

`to`	target file or directory to which the shortcut should point to.
`name`	symbolic link folder name. Default folder name is `"secure_data"`
`...`	arguments passed on to `R.utils::createLink()`

Details

A symbolic link is a special kind of file that points to another file/folder. A symbolic link does not contain the data in the target file. It simply points to another entry somewhere in the file system. This allows symbolic links to link to directories or files on remote network locations. Depending on your operating system, a link may not establish if the originating path is a network drive.

Value

NULL, Places the path or pathname to the link.

Author(s)

Daniel D. Sjoberg

See Also

`R.utils::createLink()`

Examples

```
# Using `starter_symlink()` to establish a symbolic link to a
# mapped networked data folder.
# The default name of the symlink folder is 'secure_data'
create_symlink("O:/Outcomes/Project Folder/Data")
```

`project_templates` *Project templates*

Description

The `project_templates` object defines the contents of the project templates used in `create_project()` and `use_file()`.

Usage

```
project_templates
```

Format

A named list containing the project templates.

Examples

```
if (FALSE) {
create_project(
  path = file.path(tempdir(), "Sjoberg New Project"),
  template = project_templates[["analysis"]]
)
}
```

use_project_file *Write a template file*

Description

Rather than using `create_project()` to start a new project folder, you may use `use_project_file()` to write a single file from any project template. The functions `use_project_gitignore()` and `use_project_readme()` are shortcuts for `use_project_file("gitignore")` and `use_project_file("readme")`.

Usage

```
use_project_file(
  name = NULL,
  filename = NULL,
  template = NULL,
  open = interactive()
)

use_project_gitignore(filename = NULL, template = NULL)

use_project_readme(filename = NULL, template = NULL)
```

Arguments

`name`	Name of file to write. Not sure of the files available to you? Run the function without specifying a name, and all files available within the template will print.
`filename`	Optional argument to specify the name of the file to be written. Paths/filename is relative to project base
`template`	A project template. See vignette for details.
`open`	If `TRUE`, opens the new file.

Value

NULL, places single template file in current working directory

See Also

`create_project()`

Vignette for `create_project()`

Examples

```
# only run fn interactively, will place files in current working dir
if (interactive()) {
  # create gitignore file
  use_project_file("gitignore")
  use_project_gitignore()
```

```
  # create README.md file
  use_project_file("readme")
  use_project_readme()
}
```

Index