

Package ‘org’

July 22, 2025

Title Organising Projects

Version 2024.6.5

Description A system to help you organize projects. Most analyses have three (or more) main sections: code, results, and data, each with different requirements (version control/sharing/encryption). You provide folder locations and 'org' helps you take care of the details.

Depends R (>= 3.3.0)

License MIT + file LICENSE

URL <https://www.csids.no/org/>, <https://github.com/csids/org>

BugReports <https://github.com/csids/org/issues>

Encoding UTF-8

Imports utils

Suggests testthat, knitr, rmarkdown, rstudioapi, glue

RoxygenNote 7.2.3

VignetteBuilder knitr

Date/Publication 2024-06-04 22:10:02 UTC

NeedsCompilation no

Author Richard Aubrey White [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-6747-1726>>),
CSIDS [cph]

Maintainer Richard Aubrey White <hello@rwhite.no>

Repository CRAN

Contents

create_project_quarto_external_results	2
create_project_quarto_internal_results	2
initialize_project	3
ls_files	4
move_directory	5
package_installed	5

path	6
project	6
set_results	6
write_text	7

Index	8
--------------	----------

create_project_quarto_external_results	
<i>Create example project that uses quarto with results generated outside the .qmd file</i>	

Description

Create example project that uses quarto with results generated outside the .qmd file

Usage

create_project_quarto_external_results(home, results)

Arguments

- | | |
|---------|--------------------------------------|
| home | Location of the 'home' directory. |
| results | Location of the 'results' directory. |

create_project_quarto_internal_results	
<i>Create example project that uses quarto with results generated from within the .qmd file</i>	

Description

Create example project that uses quarto with results generated from within the .qmd file

Usage

create_project_quarto_internal_results(home, results)

Arguments

- | | |
|---------|--------------------------------------|
| home | Location of the 'home' directory. |
| results | Location of the 'results' directory. |

initialize_project	<i>Initializes project</i>
--------------------	----------------------------

Description

`org::initialize_project` takes in 2+ arguments. It then saves folder locations in the return value (a new environment) and in `org::project`, which you will use in all of your subsequent code. An additional folder will be created on the user's file system (`org::project$results_today`) which corresponds to `results/YYYY-MM-DD`. The sourced folders are saved into `org::project$env`.

Usage

```
initialize_project(
  env = new.env(),
  home = NULL,
  results = NULL,
  folders_to_be_sourced = "R",
  source_folders_absolute = FALSE,
  encode_from = "UTF-8",
  encode_to = "latin1",
  ...
)
```

Arguments

<code>env</code>	The environment that the code will be sourced into (use <code>.GlobalEnv</code> to source code into the global environment)
<code>home</code>	The folder containing 'Run.R' and 'R/'
<code>results</code>	A folder inside <code>results</code> with today's date will be created and it will be accessible via <code>org::project\$results_today</code> (this is where you will store all of your results)
<code>folders_to_be_sourced</code>	The names of folders that live inside <code>home</code> and all <code>.r</code> and <code>.R</code> files inside it will be sourced into the global environment.
<code>source_folders_absolute</code>	If <code>TRUE</code> then <code>folders_to_be_sourced</code> is an absolute folder reference. If <code>FALSE</code> then <code>folders_to_be_sourced</code> is relative and inside <code>home</code> .
<code>encode_from</code>	Folders current encoding (only used on Windows)
<code>encode_to</code>	Folders final encoding (only used on Windows)
<code>...</code>	Other folders that you would like to reference

Details

For more details see the help vignette: `vignette("intro", package = "org")`

Value

Returns an environment that contains:

- Folder locations
- An environment called `env` into which the code has been sourced into. There is also a side effect where `org::project` mirrors these values.

Examples

```
org::initialize_project(
  home = paste0(tempdir(), "/git/analyses/2019/analysis3/"),
  results = paste0(tempdir(), "/dropbox/analyses_results/2019/analysis3/"),
  raw = paste0(tempdir(), "/data/analyses/2019/analysis3/")
)
org::project$results_today
org::project$raw
```

ls_files

List files and directories Equivalent to the unix ls command.

Description

List files and directories Equivalent to the unix `ls` command.

Usage

```
ls_files(path = ".", regexp = NULL)
```

Arguments

<code>path</code>	A character vector of one or more paths.
<code>regexp</code>	A regular expression that is passed to <code>list.files</code> .

Value

filepaths and directory paths as a character vector

move_directory	<i>Move directory</i>
----------------	-----------------------

Description

Move directory

Usage

```
move_directory(from, to, overwrite_to = FALSE)
```

Arguments

from	Filepath or directory path.
to	Filepath or directory path.
overwrite_to	Boolean.

package_installed	<i>Is a package installed?</i>
-------------------	--------------------------------

Description

Is a package installed?

Usage

```
package_installed(pkg)
```

Arguments

pkg	Package name.
-----	---------------

Value

Boolean. True if package is installed, false if not.

Examples

```
org::package_installed("data.table")
```

path	<i>Construct path to a file or directory</i>
------	--

Description

Construct path to a file or directory

Usage

```
path(...)
```

Arguments

... Character vectors that will be concatenated with "/" as a separator.

project	<i>Folders to be used/referenced (environment)</i>
---------	--

Description

Folders to be used/referenced (environment)

Usage

```
project
```

Format

An object of class environment of length 0.

set_results	<i>Set results folder after initialization</i>
-------------	--

Description

Set results folder after initialization

Usage

```
set_results(results, proj = org::project)
```

Arguments

results	A folder inside results with today's date will be created and it will be accessible via org::project\$results_today (this is where you will store all of your results)
proj	The project environment (default is org::project)

Value

Nothing. There is a side effect where the environments proj and org::project have the values \$results and \$results_today altered.

write_text	<i>Write text to a file</i>
------------	-----------------------------

Description

Write text to a file

Usage

```
write_text(  
  txt,  
  file = "",  
  header = "**THIS FILE IS CONSTANTLY OVERWRITTEN -- DO NOT MANUALLY EDIT**\r\n\r\n"  
)
```

Arguments

txt	Text to be written
file	File, passed through to base::cat
header	Optional header that is inserted at the top of the text file

Value

No return value.

Index

* **datasets**

project, [6](#)

create_project_quarto_external_results,
[2](#)

create_project_quarto_internal_results,
[2](#)

initialize_project, [3](#)

ls_files, [4](#)

move_directory, [5](#)

package_installed, [5](#)

path, [6](#)

project, [6](#)

set_results, [6](#)

write_text, [7](#)