Package ‘cabinets’

October 12, 2022

Title Project Specific Workspace Organization Templates

Version 0.6.0

Description Creates project specific directory and file templates that are written to a .Rprofile file. Upon starting a new R session, these templates can be used to streamline the creation of new directories that are standardized to the user's preferences and can include the initiation of a git repository, an RStudio R project, and project-local dependency management with the 'renv' package.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

URL https://github.com/nt-williams/cabinets

BugReports https://github.com/nt-williams/cabinets/issues

Imports R6, glue, rstudioapi, crayon, fs, stringr, utils, git2r, cli, renv

Suggests testthat (>= 2.1.0), withr

NeedsCompilation no

Author Nick Williams [aut, cre] (<https://orcid.org/0000-0002-1378-4831>)

Maintainer Nick Williams <ntwilliams.personal@gmail.com>

Repository CRAN

Date/Publication 2020-11-07 06:20:05 UTC

R topics documented:

`cabinets` *Streamlined organization with project specific custom templates*

Description

Cabinets makes it easy to create project specific file structure templates that can be referenced at the start of any R session. Cabinets works by writing project specific file templates to the .Rprofile file of the default working directory. Doing so allows the templates to be accessed in new R sessions without having to redefine them. On first use, users will be prompted for explicit permission to write to .Rprofile. Permission to write can be revoked at any time by setting the permission option in the .Rprofile file to FALSE. Due to these explicit permission requirements, cabinets will only work in interactive R sessions.

`cabinets_options_set` *Set cabinets options*

Description

Set cabinets options

Usage

```
cabinets_options_set(..., .envir = NULL)
```

Arguments

`...`	Options to be set
`.envir`	Environment to set options in; if NULL, will use environment at `parent.frame()`

Details

Mainly used for specifying if cabinets has permission to write to .Rprofile. Permission can be revoked at any time by opening the .Rprofile file and setting `"cabinets.permission" = FALSE`.

Value

If no options are set, returns the options specified in `options`.

`create_cabinet` *Create a cabinet template*

Description

`create_cabinet` writes code to the .Rprofile file so that when new R sessions are started, the newly created cabinet, an R6 object of class FileCabinet, is available in the global environment as a hidden object. The cabinet simply stores file location and file template information that `new_cabinet_proj` uses to create new projects with the pre-defined structure.

Usage

```
create_cabinet(name, directory, structure, .alias = name)
```

Arguments

`name`	Name of the cabinet; character of length 1. This is how the cabinet will be referenced, so best to chose something memorable.
`directory`	The file path for where the cabinet will exist.
`structure`	A list of paths of folders/files to create. See details for further explanation.
`.alias`	An optional name for the object the cabinet will be stored in R as. Defaults to `name`.

Details

Before writing to or creating a .Rprofile file, cabinets will explicitly ask for the user's permission to on exit. The cabinet structure should be defined using a list with the names defining folder paths. List values should be set to NULL.

Value

An R6 object of class FileCabinet. The code to generate this object is written to the .Rprofile file of the home directory.

See Also

`new_cabinet_proj`

`delete_cabinet` *Delete Cabinets*

Description

Delete Cabinets

Usage

```
delete_cabinet(cabinet)
```

Arguments

`cabinet`	An R6 object of class Cabinet written to the .Rprofile.

`edit_r_profile` *Open .Rprofile for editing*

Description

`edit_r_profile` opens the .Rprofile file for editing. If the .Rprofile file doesn't exist an error message will be returned. This is essentially a wrapper function for `file.edit`.

Usage

```
edit_r_profile()
```

Value

A message that .Rprofile is being opened or that it doesn't exist.

`FileCabinet` *R6 class for a cabinet*

Description

Constructs an R6 class of FileCabinet. Objects of class FileCabinet contain information that is used by `new_cabinet_proj()` to create project directories.

Public fields

`name` cabinet name.

`directory` the path to where future directories will be created, a string.

`structure` the directory structure, a list.

Methods

Public methods:

- `FileCabinet$new()`
- `FileCabinet$print()`
- `FileCabinet$clone()`

Method `new()`**:**

Usage:

```
FileCabinet$new(name, directory, structure)
```

Arguments:

`name` cabinet name.

`directory` the path to where future directories will be created, a string.

`structure` the directory structure, a list.

Details: Create a new ‘FileCabinet‘ object.

Returns: A cabinet object.

Examples:

```
FileCabinet$new("test", "a/path",
                list(code = NULL, 'data/derived' = NULL, 'data/source' = NULL))
```

Method `print()`**:**

Usage:

```
FileCabinet$print()
```

Details: Print an object of class FileCabinet.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
FileCabinet$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
## ------------------------------------------------
## Method `FileCabinet$new`
## ------------------------------------------------

FileCabinet$new("test", "a/path",
                list(code = NULL, 'data/derived' = NULL, 'data/source' = NULL))
```

get_cabinets *Print available cabinets*

Description

get_cabinets returns objects of class FileCabinet.

Usage

```
get_cabinets(envir = parent.frame())
```

Arguments

envir	The environment to check in. The default is the global environment.

Value

Objects of class FileCabinet found in the global environment.

Examples

```
get_cabinets()
```

new_cabinet_proj *Create a new project using a cabinet template*

Description

Generate new project directories using cabinet templates.

Usage

```
new_cabinet_proj(
  cabinet,
  project_name,
  r_project = TRUE,
  open = TRUE,
  renv = TRUE,
  git = TRUE,
  git_root = NULL,
  git_ignore = NULL
)
```

Arguments

`cabinet` The name of the cabinet template. Available cabinets can be found using `get_cabinets()`.

`project_name` The name of the project to store in the cabinet, a character string. Can be a file path pointing to a directory within the specified cabinet.

`r_project` Logical, should an Rproject be created. Default is TRUE if working in RStudio (only works in RStudio).

`open` Logical, if creating an Rproject, should that project be opened once created. Default is TRUE if working in RStudio (only works in RStudio).

`renv` Logical, should a `renv` project be initiated. If `TRUE`, `renv` project infrastructure will be created using `scaffold`.

`git` Logical, should a git repository be initiated.

`git_root` A path relative to the project to initiate the git repository. Default is NULL and the repository is initiated at the root of the project.

`git_ignore` Character vector of files and directories to add to .gitignore file.

Value

Creates a new directory at the path specified in the cabinet template. If `r_project` = `TRUE`, a .Rproj file will also be created using the project name. If open is set to TRUE, the new R project will opened in a new R session.

See Also

`create_cabinet`

Index