Package 'rprojtree'

July 23, 2025

Type Package

Title Create Folders and Files Structure for Data Science Projects

Version 1.0.0

Author Miguel Conde

Maintainer Miguel Conde <miguelco2000@gmail.com>

Description Use JSON templates to create folders and files structure for data science projects. Includes customized templates and accepts your own as JSON files.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.2

URL https://github.com/miguel-conde/rprojtree

Imports dplyr, magrittr, rjson, jsonlite

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-03-01 08:20:02 UTC

Contents

rprojtree-package *rprojtree*

Description

This package allows you to easily create file and directory structures for data science projects

Details

For more details see the tutorial vignette:

- `vignette("rprojtree-primer", package = "rprojtree")`

See Also

Useful links:

- https://github.com/miguel-conde/rprojtree

available_templates *available_templates*

Description

available_templates

Usage

```
available_templates()
```

Details

These are the structures defined by the currently available builtin templates:

- basic_template
  - data/
    - clean/
    - raw/
  - docs/
  - outputs/
    - files/
    - reports/
  - R/
    - global.R
    - scripts/

        * src/
    - py_template
        - data/
            * clean/
            * raw/
        - docs/
        - outputs/
            * files/
            * reports/
        - PYTHON/
            * setup.py
            * notebooks/
            * projpkj/
                · __init__.py
            * scripts/
            * tests/
    - py_r_template: a combination of basic_template and py_template

Value

A character list with the names of the built-in templates available in the package

Examples

```
available_templates()
```

---

make_prj_tree *make_prj_tree*

Description

Create a files structure from a builtin template or from a customized template provided as .json file

Usage

```
make_prj_tree(json_str, file, path = ".", verbose = FALSE)
```

Arguments

json_str	string with the name of the builtin template to use
file	string for the name of the .json file describing a customized files structure to use as template
path	string containing which directory to use as root of the files structure to create
verbose	logical, show or not info while creating the files structure

Details

- Only one of `json_str` or `file` must be supplied.
- An error is raised if any node in the .json structure is found malformed

JSON format:

- All the nodes must have a `type` and a `name`. Allowed types are `dir`, for directories, and `file` for files.
- All the directory nodes can contain any number of sub-nodes for subdirectories
- The file nodes can include a `content` field for the text the file should contain.

Value

invisible, an R list that corresponds to the JSON object in `json_str`, as returned by `rjson::fromJSON`.

See Also

`rprojtree::print_template`, `rprojtree::available_templates`, `rjson::fromJSON`

Examples

```
## Not run:
root_path = "..."
make_prj_tree(json_str = "basic_template", path = root_path)

my_template <- ".../some_template.json"
make_prj_tree(file = my_template, path = root_path)

## End(Not run)
```

`print_template` *print_template*

Description

print the files structure described by a builtin template or by a .json file

Usage

```
print_template(template_name)
```

Arguments

`template_name`	name of the builtin template or the .json file to use.

Details

`jsonlite::prettify` is internally used to add indentation to the JSON content.

Value

a JSON class object as returned by `jsonlite::prettify`.

See Also

`jsonlite::prettify`

Examples

```
print_template("basic_template")
```

Index