

Package ‘fusen’

October 13, 2022

Title Build a Package from Rmarkdown Files

Version 0.4.1

Description Use Rmarkdown First method to build your package. Start your package with documentation, functions, examples and tests in the same unique file. Everything can be set from the Rmarkdown template file provided in your project, then inflated as a package. Inflating the template copies the relevant chunks and sections in the appropriate files required for package development.

License MIT + file LICENSE

URL <https://thinkr-open.github.io/fusen/>,
<https://github.com/Thinkr-open/fusen>

Depends R (>= 3.5.0)

Imports attachment, cli, desc, devtools, glue, here (>= 1.0.0),
magrittr, parsermd (>= 0.1.0), roxygen2, stats, stringi,
tibble, tidyr, tools, usethis (>= 2.0.0), utils

Suggests knitr, pkgload, rcmdcheck, rmarkdown, rstudioapi, testthat
(>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.2.1

NeedsCompilation no

Author Sebastien Rochette [aut, cre] (<<https://orcid.org/0000-0002-1565-9313>>),
ThinkR [cph]

Maintainer Sebastien Rochette <sebastien@thinkr.fr>

Repository CRAN

Date/Publication 2022-09-29 14:50:06 UTC

R topics documented:

add_additional	2
add_dev_history	4
add_fusen_chunks	5
create_fusen	6
fill_description	7
inflate	8
load_flat_functions	9

Index	10
--------------	-----------

add_additional	<i>Add flat Rmd file that drives package development</i>
----------------	--

Description

Add flat Rmd file that drives package development

Usage

```
add_additional(
  pkg = ".",
  dev_dir = "dev",
  flat_name = "additional",
  overwrite = FALSE,
  open = TRUE
)
```

```
add_minimal(
  pkg = ".",
  dev_dir = "dev",
  flat_name = "minimal",
  overwrite = FALSE,
  open = TRUE
)
```

```
add_full(
  pkg = ".",
  dev_dir = "dev",
  flat_name = "full",
  overwrite = FALSE,
  open = TRUE
)
```

```
add_flat_template(
  template = c("full", "minimal", "additional", "teaching", "dev_history"),
  pkg = ".",
```

```

dev_dir = "dev",
flat_name = template,
overwrite = FALSE,
open = TRUE
)

```

Arguments

pkg	Path where to save file
dev_dir	Name of directory for development Rmarkdown files. Default to "dev".
flat_name	Name of the file to write in dev. Use the name of the main function of your template to get chunks pre-filled with this function name.
overwrite	Whether to overwrite existing flat Rmd template file with same name
open	Logical. Whether to open file after creation
template	Name of the template to use. See details.

Details

Choose template among the different templates available:

- "full": The full template with a reproducible package that can directly be inflated. It comes along with the "dev_history" template. Default.
- "minimal": Minimal template to start a new package when you already know fusen, along with the "dev_history" template.
- "additional": Template for an additional vignette or set of additional functions.
- "teaching": Template with a reproducible package, simpler than "full", but everything to teach the minimal structure of a package.
- "dev_history": Template with functions commonly used during package development. This does not contain chunks to write your own functions.

Abbreviated names can also be used for the different templates: "add" for additional, "min" for minimal, "teach" for teaching, "dev" for "dev_history".

add_additional(), add_minimal(), add_full() are wrapper around add_flat_template("additional"), ... However, add_dev_history() is a deprecated function from a previous version.

Value

Create flat Rmd file(s) template(s) and return its (their) path

Examples

```

# Create a new project
dummypackage <- tempfile("dummypackage")
dir.create(dummypackage)

# Add
add_flat_template(template = "teaching", pkg = dummypackage)
# Delete dummy package

```

```

unlink(dummypackage, recursive = TRUE)

# For classical use in your package
## Not run:
# first time ever using 'fusen'
add_flat_template("full")

# first time in your new package
add_flat_template("minimal")

# add new flat file for new functions
add_flat_template("add")

# add new flat template for teaching (a reduced full template)
add_flat_template("teaching")
#'
## End(Not run)

```

add_dev_history *Add dev_history.Rmd file that drives package development*

Description

add_dev_history() is deprecated. We will soon be totally switching to [add_flat_template\(\)](#), which is more consistent with development processes

add_dev_history(name = "teaching") is equivalent to `add_flat_template(template = "teaching")`

Usage

```

add_dev_history(
  pkg = ".",
  overwrite = FALSE,
  open = TRUE,
  dev_dir = "dev",
  name = c("full", "minimal", "additional", "teaching")
)

```

Arguments

pkg	Path where to save file
overwrite	Whether to overwrite existing dev_history.Rmd file
open	Logical. Whether to open file after creation
dev_dir	Name of directory for development Rmarkdown files. Default to "dev".
name	Name of the template file. See add_flat_template()

Value

Create flat Rmd file(s) and return its (their) path

Examples

```
# Create a new project
dummypackage <- tempfile("dummypackage")
dir.create(dummypackage)

# Add - Use `add_flat_template()` instead
add_flat_template(template = "teaching", pkg = dummypackage)

# Delete dummy package
unlink(dummypackage, recursive = TRUE)
```

add_fusen_chunks	<i>Add {fusen} chunks</i>
------------------	---------------------------

Description

Create {fusen} chunks inside your Rmd

Usage

```
add_fusen_chunks(
  function_name = NULL,
  export = getOption("fusen.export.functions")
)
```

Arguments

function_name	Name of the function to create. If NULL (the default), the user will be prompted to enter it.
export	Should the function be exported? Default is <code>getOption("fusen.export.functions")</code> . If NULL, the user will be prompted to enter it.

Value

A list with the context and the content, invisibly.

Examples

```
add_fusen_chunks("this")
```

create_fusen	<i>Create a new fusen project</i>
--------------	-----------------------------------

Description

Create a new fusen project

Usage

```
create_fusen(  
  path,  
  template = c("full", "minimal", "teaching"),  
  open = TRUE,  
  overwrite = FALSE,  
  with_git = FALSE  
)
```

Arguments

path	Character. Path where to create the new fusen project.
template	Character. Name of the template to be used among "full", "minimal" and "teaching".
open	Logical. Should the newly created project be opened ?
overwrite	Logical. Allow to overwrite 'dev/' files if path exists.
with_git	Logical. Should git be initialized in the newly created project ?

Details

See [add_flat_template](#) for details about the different options for template. Template "additional" is not available here as it is meant to be used with an already existing fusen.

Value

Path to dev and flat files. Side-effect: Create a new directory to build a package

Examples

```
my_path <- tempfile("mypkg")  
create_fusen(path = my_path, template = "full", open = FALSE)
```

fill_description	<i>Fill DESCRIPTION file of the package</i>
------------------	---

Description

Fill DESCRIPTION file of the package

Usage

```
fill_description(pkg = ".", fields, overwrite = FALSE)
```

Arguments

pkg	Path to package
fields	A named list of fields to add to DESCRIPTION, potentially overriding default values. See use_description for how you can set personalized defaults using package options
overwrite	Whether to overwrite existing DESCRIPTION

Value

Fill DESCRIPTION file with fields. Return path to file.

Examples

```
# Create a new project
dummypackage <- tempfile("dummypackage")
dir.create(dummypackage)

fill_description(
  pkg = dummypackage,
  fields = list(
    Title = "Build A Package From Rmarkdown file",
    Description = paste(
      "Use Rmd First method to build your package.",
      "Start your package with documentation.",
      "Everything can be set from a Rmarkdown file in your project."
    ),
    `Authors@R` = c(
      person("Sebastien", "Rochette",
        email = "sebastien@thinkr.fr",
        role = c("aut", "cre"), comment = c(ORCID = "0000-0002-1565-9313")
      ),
      person(given = "ThinkR", role = "cph")
    )
  )
)

# Delete dummy package
unlink(dummypackage, recursive = TRUE)
```

inflate	<i>Inflate Rmd to package</i>
---------	-------------------------------

Description

Inflate Rmd to package

Usage

```
inflate(
  pkg = ".",
  flat_file,
  vignette_name = "Get started",
  open_vignette = TRUE,
  check = TRUE,
  document = TRUE,
  overwrite = "ask",
  ...
)
```

Arguments

pkg	Path to package
flat_file	Path to Rmarkdown file to inflate
vignette_name	Character. Title of the resulting vignette. Use NA if you do not want to create a vignette.
open_vignette	Logical. Whether to open vignette file at the end of the process
check	Logical. Whether to check package after Rmd inflating
document	Logical. Whether to document your package using att_amend_desc
overwrite	Logical (TRUE, FALSE) or character ("ask", "yes", "no). Whether to overwrite vignette and functions if already exists.
...	Arguments passed to <code>rcmdcheck::rcmdcheck()</code> . For example, you can do <code>inflate(check = TRUE, quiet = TRUE)</code> , where <code>quiet</code> is passed to <code>rcmdcheck::rcmdcheck()</code> .

Value

Package structure. Return path to current package.

Examples

```
# Create a new project
dummypackage <- tempfile("dummypackage")
dir.create(dummypackage)

# {fusen} steps
dev_file <- add_flat_template(template = "full", pkg = dummypackage, overwrite = TRUE)
```



```
flat_file <- dev_file[grepl("flat", dev_file)]
fill_description(pkg = dummyspackage, fields = list(Title = "Dummy Package"))
inflate(
  pkg = dummyspackage, flat_file = flat_file,
  vignette_name = "Exploration of my Data", check = FALSE
)

# Explore directory of the package
# browseURL(dummyspackage)

# Try pkgdown build
# usethis::use_pkgdown()
# pkgdown::build_site(dummyspackage)
# Delete dummy package
unlink(dummyspackage, recursive = TRUE)
```

load_flat_functions *Load the code of all 'function' chunk in a flat file*

Description

Load the code of all 'function' chunk in a flat file

Usage

```
load_flat_functions(flat_file, envir = globalenv())
```

Arguments

flat_file	Path to the flat to load functions from
envir	the environment in which expr is to be evaluated. May also be NULL, a list, a data frame, a pairlist or an integer as specified to sys.call .

Value

Path to flat file loaded. Used for side effect: Load functions in the global environment.

Examples

```
## Not run:
load_flat_functions(flat_file = "dev/flat_full.Rmd")
load_flat_functions(flat_file = "dev/flat_clean_fusen_files.Rmd")

## End(Not run)
```

Index

`add_additional`, 2
`add_dev_history`, 4
`add_flat_template`, 6
`add_flat_template (add_additional)`, 2
`add_flat_template()`, 4
`add_full (add_additional)`, 2
`add_fusen_chunks`, 5
`add_minimal (add_additional)`, 2
`att_amend_desc`, 8

`create_fusen`, 6

`environment`, 9

`fill_description`, 7

`inflate`, 8

`load_flat_functions`, 9

`sys.call`, 9

`use_description`, 7