

Package ‘yamlme’

December 12, 2020

Version 0.0.0.90002

Encoding UTF-8

Title Flexible Funtion for Writing yaml Headers

Depends R(>= 3.0.0)

Imports knitr,
methods,
testthat

Suggests devtools,
distill,
roxygen2,
rmarkdown,
covr

Description This package provide a function for a flexible writing of yaml headers.
This function can be implemented in automatic generation of reports by the packages taxlist and vegetable.

LazyData true

Roxygen list(markdown = TRUE)

License GPL (>= 2)

URL <https://github.com/kamapu/yamlme>

BugReports <https://github.com/kamapu/yamlme/issues>

VignetteBuilder knitr

RoxygenNote 7.1.1.9000

R topics documented:

read_rmd	2
render_rmd	2
write_rmd	4

Index	5
--------------	----------

read_rmd

Read R-markdown Documents

Description

The function `read_rmd()` reads Rmd files into character vectors pasting a line break at the end of each line. The function `txt_body()` will do the same to one or more character vectors.

Usage

```
read_rmd(file, ..., skip_head = TRUE)
```

```
txt_body(...)
```

Arguments

<code>file</code>	Character value indicating the path and the name to the Rmd file.
<code>...</code>	Arguments passed by <code>read_rmd()</code> to <code>readLines()</code> . In <code>txt_body()</code> they are character values passed to <code>c()</code> .
<code>skip_head</code>	Logical value indicating whether the yaml head should be skip or not.

Value

A character vector that can be use as body in function `write_rmd()`.

Examples

```
## Not run:
Doc <- read_rmd(file.path(path.package("yamlme"), "taxlistjourney.Rmd"))

## End(Not run)
```

render_rmd

Render documents from object

Description

This function is a wrapper of `rmarkdown::render()` and will also work with file names but also enables the possibility of rendering from objects created by `write_rmd()`.

Usage

```
render_rmd(input, ...)

## S3 method for class 'character'
render_rmd(input, ...)

## S3 method for class 'rmd_doc'
render_rmd(
  input,
  output_file,
  temp_file = "_temp.Rmd",
  delete_temp = TRUE,
  ...
)
```

Arguments

input	Either a character value indicating the path and the name of the r-markdown file, or an object of class <code>rmd_doc</code> , written by write_rmd() .
...	Further parameters passed to rmarkdown::render() .
output_file	A character value indicating the name of the output file. It is passed to rmarkdown::render() .
temp_file	A name for a temporary file created to render the object.
delete_temp	A logical value indicating whether <code>temp_file</code> should be deleted after rendering or not.

Examples

```
## Not run:
## copy example to your working directory
filename <- "taxlistjourney.Rmd"
file.copy(from=file.path(path.package("yamlme"), filename), to=filename)

## Render the file with rmarkdown::render()
render_rmd(filename, output_file="example")

## Render the file with yamlme
text_document <- read_rmd(filename)

rmd_document <- write_rmd(title="my title", author="my name",
  output="html_document", body=text_document)

render_rmd(rmd_document, output_file="example2")

## End(Not run)
```

write_rmd	<i>Writing yaml headers</i>
-----------	-----------------------------

Description

This function generates yaml headers and Rmarkdown files by including the settings as arguments of the function. Comments and pieces of header can be also added through the argument `append`.

Usage

```
write_rmd(..., append, body = "", filename)
```

Arguments

<code>...</code>	Named arguments to be inserted in the yaml head. These arguments may be inserted either as vectors or lists.
<code>append</code>	A piece of code to be appended in the head. For instance, it can contain commented code.
<code>body</code>	The content of the document that will be inserted after the head.
<code>filename</code>	A character value with the name of the file to be written (usually a *.Rmd file). If missing, no file will be written by the function.

Value

A character vector of class `'rmd_doc'` and, by request, an Rmd file.

Index

`read_rmd`, [2](#)
`readLines()`, [2](#)
`render_rmd`, [2](#)
`rmarkdown::render()`, [2](#), [3](#)

`txt_body(read_rmd)`, [2](#)

`write_rmd`, [4](#)
`write_rmd()`, [2](#), [3](#)