Package: rdConvert (via r-universe)

August 30, 2024

Title Convert Rd files to Markdown files loaded with YAML

Version 0.0.6

Description As R6 class for converting Rd files to markdown with YAML headers. This may be useful if you wish to use package documentation in static site generators outside of the R ecosystem (e.g., React, Vue, Svelte, Gatsby, etc.). By default, Rd files are rendered into their own directory with an independent `index.md` file. The Rd name is parsed and set as the child directory name.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports dplyr, Rd2md, magrittr, cli, Rd2roxygen, R6, roxygen2md, devtools

Repository https://davidruvolo51.r-universe.dev

RemoteUrl https://github.com/davidruvolo51/rdConvert

RemoteRef HEAD

RemoteSha f95506577b41d4fb9bd27d5c6659c93683d5e3eb

Contents

`convert` *Rd to Md Workflow*

Description

'convert' is an R6 class for converting Rd files to markdown for use in static site generators. This may be useful if you wish to use non-R web frameworks for generating package documentation or if you want to add you package man files to an existing web project in markdown format. This workflow is optimized for the Gatsby JS theme 'rocket-docs', but it can be used for as a generic renderer.

To get started, create a new instance of the converter. It is also recommended to convert Rd files into an existing project.

"'r pkg <- rdConvert::convert$new() "'

Next, configure the convert to fit your project. The argument 'site_dir' is used to define the base path for the static site. It is recommended to create a nested project within your R project. 'dest_dir' is the location that the convert should use to save the markdown files. If you are using RocketDocs, this is 'src/docs/usage/' (no need to supply the site_dir here).

"'r pkg$config(project_name = "rdConvert", site_dir = "gatsby", dest_dir = "src/docs/usage/", repo_name = "rdConvert", repo_url = "https://github.com/davidruvolo51/rdConvert") "'

If 'site_dir' is not present you, 'config' will through a warning. If you are using RocketDocs, you can set the path of the 'sidebar.yml' file via the 'sidebar_yml_path' argument. By default the location is 'src/config/sidebar.yml'. You can adjust the configuration at any point.

Next, create the entry and output points. This will collate all 'man' files and generate the output path using the 'site_dir' and 'dest_dir' arguments defined in by the '$config' function. If you are using RocketDocs, the sidebar data 'labels' and 'links' are generated automatically.

"'r pkg$set_entries() # build pkg$entries # view "'

Once the entry and output points are created, run '$set_destinations'. This function will verify that all directories in a path exist. If a location does not exist, the folder will be created.

"'r pkg$set_destinations() "'

The returned md file may not always be perfect. There may be extra spaces and multiple blank lines. Table layouts may not render appropriately. The function '$format_markdown()' provides some surface-level formatting for post-converted markdown files.

"'r pkg$format_markdown() "'

If you want to generate YAML for the files, use the '$add_yaml()' function. This function isn't perfect. If you are using RocketDocs it will cause the title to render twice as it was rendered via '$convert_rds'. I'm still working on this feature.

"'r pkg$add_yaml() "'

If you are using RocketDocs, I've included two methods for generating the 'sidebar.yml'. 'set_sidebar_yml' will generate the markup using the '$entries' object. The yml configuration file can be saved using the '$save_sidebar_yml' function.

"'r pkg$set_sidebar_yml() pkg$save_sidebar_yml() "'

Tips for Success

This workflow isn't perfect, but it will help provide a basic workflow for exporting Rd files for use in non-R web projects. Here are some tips for success.

- **Create static site project first**: It is a good idea to create the static site before you convert the Rd files. This will ensure the output paths are initiated before you build the paths. - **Turn of Markdown support**: While experimenting with this workflow, I noticed that roxygen2 was rendering markdown. To display this, use '@noMd' in your R files or disable it package wide in the 'DESCRIPTION » Roxygen: list(markdown = TRUE)'.

During active development, rebuilding Rd files and rerunning the steps outlined above, can be a bit tidious. To make this process a bit easier, run the method '$rebuild'. This will run 'devtools::check_man' and '$convert_rds' and '$format_markdown'. Set 'set_entries' to 'TRUE' if you have the Rd files were added or removed. You can also auto add YAMLs by setting 'add_yaml' to TRUE.

"'r pkg$rebuild(set_entries = TRUE, add_yaml = FALSE) "'

Value

Convert Rd files to Markdown files

Public fields

`site_dir` location of the of the static site directory

`entry_dir` path to Rd files

`dest_dir` output directory

`entries` list of entry files and output paths

`file_format` save file as md or mdx

`yml` configuration file for sidebar.yml

`repo_name` name of the repo (e.g., Github, GitLab, etc.)

`repo_url` address repo (e.g., GitHub, GitLab, etc.)

`sidebar_yml_path` path to RocketDocs sidebar.yml

`ignore` Rd files to ignore

Methods

Public methods:

- `convert$config()`
- `convert$set_entries()`
- `convert$set_destinations()`
- `convert$convert_rds()`
- `convert$format_markdown()`
- `convert$add_yaml()`
- `convert$set_sidebar_yml()`
- `convert$save_sidebar_yml()`

- convert$rebuild()
- convert$clone()

Method `config()`: Configure a new converter

Usage:

```
convert$config(
  site_dir = "gatsby",
  dest_dir = "src/docs/usage",
  file_format = "md",
  repo_name,
  repo_url,
  sidebar_yml_path = "src/config/sidebar.yml",
  ignore = NULL
)
```

Arguments:

`site_dir` location of the static site directory

`dest_dir` output directory for markdown files (excl. site dir)

`file_format` save file as md or mdx

`repo_name` name of the repo (e.g., Github, GitLab, etc.)

`repo_url` address repo (e.g., GitHub, GitLab, etc.)

`sidebar_yml_path` path to sidebar yml config (excl. site dir)

`ignore` man files to ignore

Method `set_entries()`: Set entry points and output file paths

Usage:

```
convert$set_entries()
```

Method `set_destinations()`: set destinitions

Usage:

```
convert$set_destinations()
```

Method `convert_rds()`: batch convert Rd files to markdown files with YAML

Usage:

```
convert$convert_rds()
```

Method `format_markdown()`: format markdown files

Usage:

```
convert$format_markdown()
```

Method `add_yaml()`: inject YAML from Rd files into rendered markdown files

Usage:

```
convert$add_yaml()
```

Method `set_sidebar_yml()`: generate sidebar yml

Usage:

```
convert$set_sidebar_yml()
```

Arguments:

`home` display home link in sidebar

Method `save_sidebar_yml()`**:** write yml config

Usage:

```
convert$save_sidebar_yml()
```

Method `rebuild()`**:** rebuild R man files and convert to markdown

Usage:

```
convert$rebuild(entries = FALSE, yaml = FALSE)
```

Arguments:

`entries` If TRUE, entry and points will be rebuilt

`yaml` If TRUE, yamls will be updated

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

Usage:

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

Arguments:

`deep` Whether to make a deep clone.

Author(s)

dcruvolo

`create_destinations` *Create Output Points*

Description

Take a string containing a file path and validate the existence of all paths and create directories where applicable.

Usage

```
create_destinations(x)
```

Arguments

`x`	a string containing a relative file path

format *Formatter*

Description

Additional functions for formatting strings and Rd ojbects

Usage

```
format(x)
```

Arguments

x	an object

format_blank_lines *Format Sequential blank lines*

Description

In some instances, there may be more than one blank line in between paragraphs. This function ensures there is only 1 blank line.

“‘r format_blank_lines(c("This has", "", "", "", "multiple", "", "","", "", "blank lines")) “‘

Usage

```
format_blank_lines(x)
```

Arguments

x	a character

format_leading_whitespace *Format Leading Whitespace*

Description

In many text blocks, there a leading white space is added. This function removes it!

Usage

```
format_leading_whitespace(x)
```

Arguments

x	a string from an md file

Examples

```
format_leading_whitespace("    This is a cool string!   ")
```

format_md_table *Format Markdown Table*

Description

In converted markdown files, the markup can be a bit funny and will render properly. This is caused by extra spaces, no spaces between column characters, and multiple backticks around arguments. This function cleans markdown tables.

Usage

```
format_md_table(x)
```

Arguments

x	a string or array containing markdown

format_path *Format Path*

Description

Remove trailing forward slash in file paths.

Usage

```
format_path(x)
```

Arguments

x	a string containing a file path

Examples

```
format_path("path/to/doc/")
```

format_rd_path *Format Rd File name*

Description

Remove the file extension from Rd files.

Usage

```
format_rd_path(x)
```

Arguments

x	a file name

Examples

```
format_rd_path("man/myfile.Rd")
```

format_yaml_keywords *Format YAML keywords*

Description

Format Rd keywords (post-conversion), into a JS formatted string.

Usage

```
format_yaml_keywords(x)
```

Arguments

x	an Rd keyword array

Examples

```
format_yaml_keywords(c("I", "think", "R", "is", "cool"))
```

format_yaml_text *Format YAML Text*

Description

Converted Rd files may not always render YAML markdown. This function rendered Rd into Markdown and ensures YAML is quoted.

Usage

```
format_yaml_text(x)
```

Arguments

x	an Rd string

Examples

```
format_yaml_text("This is a \\bold{cool} string")
```

Index