

# Package ‘altdoc’

January 23, 2025

**Type** Package

**Title** Package Documentation Websites with 'Quarto', 'Docsify',  
'Docute', or 'MkDocs'

**Version** 0.5.0

**Description** This is a simple and powerful package to create, render, preview,  
and deploy documentation websites for 'R' packages. It is a lightweight and  
flexible alternative to 'pkgdown', with support for many documentation  
generators, including 'Quarto', 'Docute', 'Docsify', and 'MkDocs'.

**License** MIT + file LICENSE

**URL** <https://altdoc.etiennebacher.com>,  
<https://github.com/etiennebacher/altdoc>

**BugReports** <https://github.com/etiennebacher/altdoc/issues>

**Imports** cli, desc, evaluate, fs, quarto, tools, utils

**Suggests** covr, digest, downlit, future, future.apply, knitr, servr,  
spelling, testthat (>= 3.0.0), usethis, withr, yaml

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Language** en-US

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Etienne Bacher [aut, cre, cph],  
Vincent Arel-Bundock [aut] (<<https://orcid.org/0000-0003-2042-7063>>)

**Maintainer** Etienne Bacher <etienne.bacher@protonmail.com>

**Repository** CRAN

**Date/Publication** 2025-01-23 16:40:07 UTC

## Contents

preview_docs . . . . .	2
render_docs . . . . .	3
setup_docs . . . . .	5
setup_github_actions . . . . .	7
<b>Index</b>	<b>8</b>

---

preview_docs	<i>Preview the documentation in a webpage or in viewer</i>
--------------	------------------------------------------------------------

---

### Description

Preview the documentation in a webpage or in viewer

### Usage

```
preview_docs(path = ".")
```

### Arguments

path                    Path to the package root directory.

### Value

No value returned. If RStudio is used, it shows a site preview in Viewer. To preview the site in a browser or in another text editor (ex: VS Code), see the vignette on the altdoc website.

### Examples

```
if (interactive()) {
  preview_docs()
}

# This is an example to illustrate that code-generated images are properly
# displayed. See the `altdoc` website for a rendered version.
with(mtcars, plot(mpg, wt))
```

---

render_docs	<i>Update documentation</i>
-------------	-----------------------------

---

### Description

Render and update the function reference manual, vignettes, README, NEWS, CHANGELOG, LICENSE, and CODE\_OF\_CONDUCT sections, if they exist. This function overwrites the content of the 'docs/' folder. See details below.

### Usage

```
render_docs(path = ".", verbose = FALSE, parallel = FALSE, freeze = FALSE, ...)
```

### Arguments

path	Path to the package root directory.
verbose	Logical. Print Rmarkdown or Quarto rendering output.
parallel	Logical. Render man pages and vignettes in parallel using the future framework. In addition to setting this argument to TRUE, users must define the parallelism plan in future. See the examples section below.
freeze	Logical. If TRUE and a man page or vignette has not changed since the last call to render_docs(), that file is skipped. File hashes are stored in altdoc/freeze.rds. If that file is deleted, all man pages and vignettes will be rendered anew.
...	Additional arguments are ignored.

### Details

This function searches the root directory and the inst/ directory for specific filenames, renders/converts/copies them to the docs/ directory. The order of priority for each file is established as follows:

- docs/README.md
  - README.md, README.qmd, README.Rmd
- docs/NEWS.md
  - NEWS.md, NEWS.txt, NEWS, NEWS.Rd
  - Note: Where possible, Github contributors and issues are linked automatically.
- docs/CHANGELOG.md
  - CHANGELOG.md, CHANGELOG.txt, CHANGELOG
- docs/CODE\_OF\_CONDUCT.md
  - CODE\_OF\_CONDUCT.md, CODE\_OF\_CONDUCT.txt, CODE\_OF\_CONDUCT
- docs/LICENSE.md
  - LICENSE.md, LICENSE.txt, LICENSE
- docs/LICENCE.md
  - LICENCE.md, LICENCE.txt, LICENCE

### Altdoc variables

The settings files in the `altdoc/` directory can include `$ALTDOC` variables which are replaced automatically by `altdoc` when calling `render_docs()`:

- `$ALTDOC_PACKAGE_NAME`: Name of the package from DESCRIPTION.
- `$ALTDOC_PACKAGE_VERSION`: Version number of the package from DESCRIPTION
- `$ALTDOC_PACKAGE_URL`: First URL listed in the DESCRIPTION file of the package.
- `$ALTDOC_PACKAGE_URL_GITHUB`: First URL that contains "github.com" from the URLs listed in the DESCRIPTION file of the package. If no such URL is found, lines containing this variable are removed from the settings file.
- `$ALTDOC_MAN_BLOCK`: Nested list of links to the individual help pages for each exported function of the package. The format of this block depends on the documentation generator.
- `$ALTDOC_VIGNETTE_BLOCK`: Nested list of links to the vignettes. The format of this block depends on the documentation generator.
- `$ALTDOC_VERSION`: Version number of the altdoc package.

Also note that you can store images and static files in the `altdoc/` directory. All the files in this folder are copied to `docs/` and made available in the root of the website, so you can link to them easily.

### Altdoc preambles

When you call `render_docs()`, `altdoc` will automatically paste the content of one of these three files to the top of a document:

- `altdoc/preamble_vignettes_qmd.yml`
- `altdoc/preamble_vignettes_rmd.yml`
- `altdoc/preamble_man_qmd.yml`

The README file uses the vignette preamble.

To preempt this behavior, add your own preamble to the README file or to a vignette.

### Freeze

When working on a package, running `render_docs()` to preview changes can be a time-consuming road block. The argument `freeze = TRUE` tries to improve the experience by preventing rerendering of files that have not changed since the last time `render_docs()` was ran. Note that changes to package internals will not cause a rerender, so rerendering the entire docs can still be necessary.

For non-Quarto formats, this is done by creating a `freeze.rds` file in `altdoc/` that is able to determine which documentation files have changed.

For the Quarto format, we rely on the [Quarto freeze](#) feature. Freezing a document needs to be set either at a project or per-file level. Freezing a document needs to be set either at a project or per-file level. To do so, add to either `quarto_website.yml` or the frontmatter of a file:

```
execute:
  freeze: auto
```

### Auto-link for Quarto websites

When the `code-link` format setting is `true` in `altdoc/quarto_website.yml` and the `downlit` package is installed, `altdoc` will use the `downlit` package to replace the function names on the package website by links to web-based package documentation. This works for base R libraries and any package published on CRAN.

To allow internal links to functions documented by `altdoc`, we need to include links to correct URLs in the `altdoc/pkgdown.yml` file. By default, this file is populated with links to the first URL in the `DESCRIPTION`.

Importantly, `downlit` requires the `pkgdown.yml` file to be live on the website to create links. This means that links will generally not be updated when making purely local changes. Also, links may not be updated the first time an `altdoc` website is published to the web.

### Examples

```
if (interactive()) {
  render_docs()

  # parallel rendering
  library(future)
  plan(multicore)
  render_docs(parallel = TRUE)
}
```

---

setup\_docs

*Initialize documentation website settings*

---

### Description

Creates a subdirectory called `altdoc/` in the package root directory to store the settings files used to by one of the documentation generator utilities (`docsify`, `docute`, `mkdocs`, or `quarto_website`). The files in this folder are never altered automatically by `altdoc` unless the user explicitly calls `overwrite=TRUE`. They can thus be edited manually to customize the sidebar and website.

### Usage

```
setup_docs(tool, path = ".", overwrite = FALSE)
```

### Arguments

<code>tool</code>	String. "docsify", "docute", "mkdocs", or "quarto_website".
<code>path</code>	Path to the package root directory.
<code>overwrite</code>	Logical. If <code>TRUE</code> , overwrite existing files. Warning: This will completely delete the settings files in the <code>altdoc</code> directory, including any customizations you may have made.

## Package structure

altdoc makes assumptions about your package structure:

- The homepage of the website is stored in `README.qmd`, `README.Rmd`, or `README.md`.
- `vignettes/` stores the vignettes in `.md`, `.Rmd` or `.qmd` format.
- `docs/` stores the rendered website. This folder is overwritten every time a user calls `render_docs()`, so you should not edit it manually.
- `altdoc/` stores the settings files created by `setup_docs()`. These files are never modified automatically after initialization, so you can edit them manually to customize the settings of your documentation and website. All the files stored in `altdoc/` are copied to `docs/` and made available as static files in the root of the website.
- These files are imported automatically: `NEWS.md`, `CHANGELOG.md`, `CODE_OF_CONDUCT.md`, `LICENSE.md`, `LICENCE.md`.

## Altdoc variables

The settings files in the `altdoc/` directory can include `$ALTDOC` variables which are replaced automatically by `altdoc` when calling `render_docs()`:

- `$ALTDOC_PACKAGE_NAME`: Name of the package from `DESCRIPTION`.
- `$ALTDOC_PACKAGE_VERSION`: Version number of the package from `DESCRIPTION`
- `$ALTDOC_PACKAGE_URL`: First URL listed in the `DESCRIPTION` file of the package.
- `$ALTDOC_PACKAGE_URL_GITHUB`: First URL that contains "github.com" from the URLs listed in the `DESCRIPTION` file of the package. If no such URL is found, lines containing this variable are removed from the settings file.
- `$ALTDOC_MAN_BLOCK`: Nested list of links to the individual help pages for each exported function of the package. The format of this block depends on the documentation generator.
- `$ALTDOC_VIGNETTE_BLOCK`: Nested list of links to the vignettes. The format of this block depends on the documentation generator.
- `$ALTDOC_VERSION`: Version number of the `altdoc` package.

Also note that you can store images and static files in the `altdoc/` directory. All the files in this folder are copied to `docs/` and made available in the root of the website, so you can link to them easily.

## Altdoc preambles

When you call `render_docs()`, `altdoc` will automatically paste the content of one of these three files to the top of a document:

- `altdoc/preamble_vignettes_qmd.yml`
- `altdoc/preamble_vignettes_rmd.yml`
- `altdoc/preamble_man_qmd.yml`

The `README` file uses the vignette preamble.

To preempt this behavior, add your own preamble to the `README` file or to a vignette.

## Examples

```
if (interactive()) {  
  
  # Create docute documentation  
  setup_docs(tool = "docute")  
  
  # Create docsify documentation  
  setup_docs(tool = "docsify")  
  
  # Create mkdocs documentation  
  setup_docs(tool = "mkdocs")  
  
  # Create quarto website documentation  
  setup_docs(tool = "quarto_website")  
}
```

---

setup\_github\_actions *Create a Github Actions workflow*

---

## Description

This function creates a Github Actions workflow in ".github/workflows/altdoc.yaml". This workflow will automatically render the website using the setup specified in the folder "altdoc" and will push the output to the branch "gh-pages".

## Usage

```
setup_github_actions(path = ".")
```

## Arguments

path                    Path to the package root directory.

## Value

No value returned. Creates the file ".github/workflows/altdoc.yaml"

## Examples

```
if (interactive()) {  
  setup_github_actions()  
}
```

# Index

`preview_docs`, [2](#)

`render_docs`, [3](#)

`setup_docs`, [5](#)

`setup_github_actions`, [7](#)