

# Package ‘quarto’

May 18, 2021

**Title** R Interface to 'Quarto' Markdown Publishing System

**Version** 0.3

**Description** Convert R Markdown documents and 'Jupyter' notebooks to a variety of output formats using 'Quarto'.

**Imports** utils,rmarkdown,jsonlite,yaml,processx,rstudioapi,later

**Suggests** testthat, rsconnect

**SystemRequirements** Quarto command line tools  
(<https://github.com/quarto-dev/quarto-cli>).

**License** GPL (>= 2)

**URL** <https://github.com/quarto-dev/quarto-r>

**BugReports** <https://github.com/quarto-dev/quarto-r/issues>

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** JJ Allaire [aut, cre] (<<https://orcid.org/0000-0003-0174-9868>>)

**Maintainer** JJ Allaire <jj@rstudio.com>

**Repository** CRAN

**Date/Publication** 2021-05-18 16:50:02 UTC

## R topics documented:

quarto . . . . .	2
quarto_metadata . . . . .	2
quarto_path . . . . .	3
quarto_publish . . . . .	3
quarto_render . . . . .	4
quarto_serve . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

quarto	<i>Internal package state</i>
--------	-------------------------------

---

**Description**

Internal package state

**Usage**

```
quarto
```

**Format**

An object of class environment of length 0.

---

quarto_metadata	<i>Read Quarto Metadata</i>
-----------------	-----------------------------

---

**Description**

Read YAML metadata for an input file or project.

**Usage**

```
quarto_metadata(input = ".")
```

**Arguments**

input            The input file or project directory to read metadata for.

**Value**

Named list with metadata. For input files, the named list is keyed by output format. For projects, all project level metadata defined in `_quarto` is contained in the list.

**Examples**

```
## Not run:  
# Read metadata for file  
quarto_metadata("notebook.Rmd")  
  
# Read metadata for project  
quarto_metadata("myproject")  
  
## End(Not run)
```

---

quarto_path	<i>Path to the quarto binary</i>
-------------	----------------------------------

---

**Description**

Determine the path to the quarto binary. Uses QUARTO\_PATH environment variable if defined, otherwise uses `Sys.which()`.

**Usage**

```
quarto_path()
```

**Value**

Path to quarto binary (or NULL if not found)

---

quarto_publish	<i>Publish to RStudio Connect</i>
----------------	-----------------------------------

---

**Description**

Publish a quarto document or project to RStudio Connect

**Usage**

```
quarto_publish(  
  input = ".",  
  name = NULL,  
  method = c("rsconnect"),  
  server = NULL,  
  account = NULL,  
  render = TRUE,  
  launch_browser = interactive()  
)
```

**Arguments**

input	The input file or project directory to be published. Defaults to current working directory.
name	Name for publishing (names must be unique within an account). For projects, defaults to the name provided by the project (alternatively uses the base name of the input).
method	Publishing method (currently only "rsconnect" is available)
server	Server name. Required only if you use the same account name on multiple servers.

account	Account to deploy to. This parameter is only required for the initial deployment when there are multiple accounts configured on the system.
render	TRUE to render locally before publishing.
launch_browser	If TRUE, the system's default web browser will be launched automatically after deployment. Defaults to TRUE in interactive sessions only.

### Examples

```
## Not run:
library(quarto)
quarto_publish()

## End(Not run)
```

---

quarto_render	<i>Render Markdown</i>
---------------	------------------------

---

### Description

Render the input file to the specified output format using quarto. If the input requires computations (e.g. for Rmd or Jupyter files) then those computations are performed before rendering.

### Usage

```
quarto_render(
  input = NULL,
  output_format = NULL,
  output_file = NULL,
  execute = TRUE,
  execute_params = NULL,
  execute_dir = NULL,
  cache = NULL,
  cache_refresh = FALSE,
  kernel_keepalive = NULL,
  kernel_restart = FALSE,
  debug = FALSE,
  quiet = FALSE,
  pandoc_args = NULL
)
```

### Arguments

input	The input file or project directory to be rendered (defaults to rendering the project in the current working directory).
output_format	Target output format (defaults to "html"). The option "all" will render all formats defined within the file or project.

output_file	The name of the output file. If using NULL then the output filename will be based on filename for the input file.
execute	Whether to execute embedded code chunks.
execute_params	A list of named parameters that override custom params specified within the YAML front-matter.
execute_dir	The working directory in which to execute embedded code chunks.
cache	Cache execution output (uses knitr cache and jupyter-cache respectively for Rmd and Jupyter input files).
cache_refresh	Force refresh of execution cache.
kernel_keepalive	Keep Jupyter kernel alive (defaults to 300 seconds). Note this option is only applicable for rendering Jupyter notebooks or Jupyter markdown.
kernel_restart	Restart keepalive Jupyter kernel before render. Note this option is only applicable for rendering Jupyter notebooks or Jupyter markdown.
debug	Leave intermediate files in place after render.
quiet	Suppress warning and other messages.
pandoc_args	Additional command line options to pass to pandoc.

### Examples

```
## Not run:
# Render R Markdown
quarto_render("notebook.Rmd")
quarto_render("notebook.Rmd", output_format = "pdf")

# Render Jupyter Notebook
quarto_render("notebook.ipynb")

# Render Jupyter Markdown
quarto_render("notebook.md")

## End(Not run)
```

---

quarto\_serve

*Quarto Development Server*


---

### Description

Run a local web server for a Quarto project.

**Usage**

```
quarto_serve(  
  dir = NULL,  
  port = "auto",  
  browse = TRUE,  
  watch = TRUE,  
  navigate = TRUE  
)  
  
quarto_serve_stop()
```

**Arguments**

dir	The project directory to serve (defaults to current working directory)
port	Port to listen on (defaults to 4848)
browse	Open a browser to preview the site. Defaults to using the RStudio Viewer when running within RStudio. Pass a function (e.g. <code>utils::browseURL</code> to override this behavior).
watch	Watch for changes and automatically reload browser.
navigate	Automatically navigate the preview browser to the most recently rendered document.

**Examples**

```
## Not run:  
# Run a local server for the project in the current directory  
quarto_serve()  
  
# Run server for project in "myproj" directory and preview in external  
# browser (rather than RStudio Viewer)  
quarto_serve("myproj", open = utils::browseURL)  
  
# Stop any running quarto server  
quarto_serve_stop()  
  
## End(Not run)
```

# Index

## \* datasets

quarto, [2](#)

quarto, [2](#)

quarto\_metadata, [2](#)

quarto\_path, [3](#)

quarto\_publish, [3](#)

quarto\_render, [4](#)

quarto\_serve, [5](#)

quarto\_serve\_stop (quarto\_serve), [5](#)