

Package ‘quarto’

March 6, 2024

Title R Interface to 'Quarto' Markdown Publishing System

Version 1.4

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

License GPL (>= 2)

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

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

Depends R (>= 3.6)

Imports cli, jsonlite, later, processx, rlang, rmarkdown, rstudioapi,
tools, utils, yaml

Suggests curl, knitr, rsconnect (>= 0.8.26), testthat (>= 3.1.7),
withr, xfun

VignetteBuilder quarto

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

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

NeedsCompilation no

Author JJ Allaire [aut] (<<https://orcid.org/0000-0003-0174-9868>>),
Christophe Dervieux [cre, aut]
(<<https://orcid.org/0000-0003-4474-2498>>),
Posit Software, PBC [cph, fnd]

Maintainer Christophe Dervieux <cderv@posit.co>

Repository CRAN

Date/Publication 2024-03-06 15:40:02 UTC

R topics documented:

is_using_quarto	2
quarto_add_extension	3
quarto_binary_sitrep	4
quarto_create_project	4
quarto_inspect	5
quarto_path	6
quarto_preview	7
quarto_publish_doc	8
quarto_render	10
quarto_serve	12
quarto_use_template	12
quarto_version	13
Index	14

is_using_quarto	<i>Check if a directory is using quarto</i>
-----------------	---

Description

This function will check if a directory is using quarto by looking for

- `_quarto.yml` at its root
- at least one `.qmd` file in the directory

Usage

```
is_using_quarto(dir = ".", verbose = FALSE)
```

Arguments

<code>dir</code>	The directory to check
<code>verbose</code>	print message about the result of the check

Examples

```
dir.create(tmpdir <- tempfile())
is_using_quarto(tmpdir)
file.create(file.path(tmpdir, "_quarto.yml"))
is_using_quarto(tmpdir)
unlink(tmpdir, recursive = TRUE)
```

quarto_add_extension *Install a Quarto extensions*

Description

Add an extension to this folder or project by running `quarto add`

Usage

```
quarto_add_extension(  
  extension = NULL,  
  no_prompt = FALSE,  
  quiet = FALSE,  
  quarto_args = NULL  
)
```

Arguments

<code>extension</code>	The extension to install, either an archive or a GitHub repository as described in the documentation https://quarto.org/docs/extensions/managing.html .
<code>no_prompt</code>	Do not prompt to confirm approval to download external extension.
<code>quiet</code>	Suppress warning and other messages.
<code>quarto_args</code>	Character vector of other quarto CLI flag pass to the command. This is mainly for advanced usage, e.g it can be useful for new options added to quarto CLI and not yet supported as function argument.

Extension Trust

Quarto extensions may execute code when documents are rendered. Therefore, if you do not trust the author of an extension, we recommend that you do not install or use the extension. By default `no_prompt = FALSE` which means that the function will ask for explicit approval when used interactively, or disallow installation.

Examples

```
## Not run:  
# Install a template and set up a draft document from a GitHub repository  
quarto_add_extension("quarto-ext/fontawesome")  
  
# Install a template and set up a draft document from a ZIP archive  
quarto_add_extension("https://github.com/quarto-ext/fontawesome/archive/refs/heads/main.zip")  
  
## End(Not run)
```

quarto_binary_sitrep *Check configurations for quarto binary used*

Description

This function check the configuration for the quarto package R package to detect a possible difference with version used by RStudio IDE.

Usage

```
quarto_binary_sitrep(verbose = TRUE, debug = FALSE)
```

Arguments

verbose if FALSE, only return the result of the check.
debug if TRUE, print more information about value set in configurations.

Value

TRUE if this package should be using the same quarto binary as the RStudio IDE. FALSE otherwise if a difference is detected or quarto is not found. Use verbose = TRUE or debug = TRUE to get detailed information.

Examples

```
quarto_binary_sitrep(verbose = FALSE)  
quarto_binary_sitrep(verbose = TRUE)  
quarto_binary_sitrep(debug = TRUE)
```

quarto_create_project *Create a quarto project*

Description

This function calls `quarto create project <type> <name>`. It will create a new directory with the project name and add some skeletons files for the project type chosen.

Usage

```
quarto_create_project(  
  name,  
  type = "default",  
  dir = ".",  
  no_prompt = FALSE,  
  quiet = FALSE,  
  quarto_args = NULL  
)
```

Arguments

name	The name of the project and the directory that will be created.
type	The type of project to create. As of 1.4, it can be one of default, website, blog, book, manuscript, confluence.
dir	The directory where to create the new Quarto project.
no_prompt	Do not prompt to confirm approval to download external extension.
quiet	Suppress warning and other messages.
quarto_args	Character vector of other quarto CLI flag pass to the command. This is mainly for advanced usage, e.g it can be useful for new options added to quarto CLI and not yet supported as function argument.

Quarto version required

This function require Quarto 1.4 or higher. Use `quarto_version()` to check the version of Quarto detected.

See Also

Quarto documentation on [Quarto projects](#)

quarto_inspect	<i>Inspect Quarto Input File or Project</i>
----------------	---

Description

Inspect a Quarto project or input path. Inspecting a project returns its config and engines. Inspecting an input path return its formats, engine, and dependent resources.

Usage

```
quarto_inspect(input = ".", profile = NULL, quiet = FALSE, quarto_args = NULL)
```

Arguments

input	The input file or project directory to inspect.
profile	Quarto project profile(s) to use. Either a character vector of profile names or NULL to use the default profile.
quiet	Suppress warning and other messages.
quarto_args	Character vector of other quarto CLI flag pass to the command. This is mainly for advanced usage, e.g it can be useful for new options added to quarto CLI and not yet supported as function argument.

Value

Named list. For input files, the list contains the elements `quarto`, `engines`, `formats`, `resources`, plus `project` if the file is part of a Quarto project. For projects, the list contains the elements `quarto`, `dir`, `engines`, `config` and `files`.

Examples

```
## Not run:  
# Inspect input file file  
quarto_inspect("notebook.Rmd")  
  
# Inspect project  
quarto_inspect("myproject")  
  
# Inspect project's advanced profile  
quarto_inspect(  
  input = "myproject",  
  profile = "advanced"  
)  
  
## End(Not run)
```

`quarto_path`*Path to the quarto binary*

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)

Description

Render and preview a Quarto document or website project.

Usage

```
quarto_preview(  
  file = NULL,  
  render = "auto",  
  port = "auto",  
  host = "127.0.0.1",  
  browse = TRUE,  
  watch = TRUE,  
  navigate = TRUE  
)  
  
quarto_preview_stop()
```

Arguments

file	The document or website project directory to preview (defaults to current working directory)
render	For website preview, the most recent execution results of computational documents are used to render the site (this is to optimize startup time). If you want to perform a full render prior to serving pass "all" or a vector of specific formats to render. Pass "default" to render the default format for the site. For document preview, the document is rendered prior to preview (pass FALSE to override this).
port	Port to listen on (defaults to 4848)
host	Hostname to bind to (defaults to 127.0.0.1)
browse	Open a browser to preview the content. 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.

Details

Automatically reloads the browser when input files are re-rendered or document resources (e.g. CSS) change.

Examples

```
## Not run:
# Preview the project in the current directory
quarto_preview()

# Preview a document
quarto_preview("document.qmd")

# Preview the project in "myproj" directory and use external browser
# (rather than RStudio Viewer)
quarto_preview("myproj", open = utils::browseURL)

# Stop any running quarto preview
quarto_preview_stop()

## End(Not run)
```

quarto_publish_doc *Publish Quarto Documents*

Description

Publish Quarto documents to Posit Connect, ShinyApps, and RPubs

Usage

```
quarto_publish_doc(
  input,
  name = NULL,
  title = NULL,
  server = NULL,
  account = NULL,
  render = c("local", "server", "none"),
  metadata = list(),
  ...
)

quarto_publish_app(
  input = getwd(),
  name = NULL,
  title = NULL,
  server = NULL,
  account = NULL,
  render = c("local", "server", "none"),
  metadata = list(),
  ...
)
```



```

quarto_publish_site(
  input = getwd(),
  name = NULL,
  title = NULL,
  server = NULL,
  account = NULL,
  render = c("local", "server", "none"),
  metadata = list(),
  ...
)

```

Arguments

<code>input</code>	The input file or project directory to be published. Defaults to the current working directory.
<code>name</code>	Name for publishing (names must be unique within an account). Defaults to the name of the input.
<code>title</code>	Free-form descriptive title of application. Optional; if supplied, will often be displayed in favor of the name. When deploying a new document, you may supply only the title to receive an auto-generated name
<code>account, server</code>	Uniquely identify a remote server with either your user account, the server name, or both. If neither are supplied, and there are multiple options, you'll be prompted to pick one. Use accounts() to see the full list of available options.
<code>render</code>	local to render locally before publishing; server to render on the server; none to use whatever rendered content currently exists locally. (defaults to local)
<code>metadata</code>	Additional metadata fields to save with the deployment record. These fields will be returned on subsequent calls to deployments() . Multi-value fields are recorded as comma-separated values and returned in that form. Custom value serialization is the responsibility of the caller.
<code>...</code>	Named parameters to pass along to <code>rsconnect::deployApp()</code>

Examples

```

## Not run:
library(quarto)
quarto_publish_doc("mydoc.qmd")
quarto_publish_app(server = "shinyapps.io")
quarto_publish_site(server = "rstudioconnect.example.com")

## 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,
  execute_daemon = NULL,
  execute_daemon_restart = FALSE,
  execute_debug = FALSE,
  use_freezer = FALSE,
  cache = NULL,
  cache_refresh = FALSE,
  metadata = NULL,
  metadata_file = NULL,
  debug = FALSE,
  quiet = FALSE,
  profile = NULL,
  quarto_args = NULL,
  pandoc_args = NULL,
  as_job = getOption("quarto.render_as_job", "auto")
)
```

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, the output filename will be based on the filename for the input file. output_file is mapped to the --output option flag of the quarto CLI. It is expected to be a filename only, not a path, relative or absolute.
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.
execute_daemon	Keep Jupyter kernel alive (defaults to 300 seconds). Note this option is only applicable for rendering Jupyter notebooks or Jupyter markdown.
execute_daemon_restart	Restart keepalive Jupyter kernel before render. Note this option is only applicable for rendering Jupyter notebooks or Jupyter markdown.
execute_debug	Show debug output for Jupyter kernel.
use_freezer	Force use of frozen computations for an incremental file render.
cache	Cache execution output (uses knitr cache and jupyter-cache respectively for Rmd and Jupyter input files).
cache_refresh	Force refresh of execution cache.
metadata	An optional named list used to override YAML metadata. It will be passed as a YAML file to <code>--metadata-file</code> CLI flag. This will be merged over <code>metadata-file</code> options if both are specified.
metadata_file	A yaml file passed to <code>--metadata-file</code> CLI flags to override metadata. This will be merged with <code>metadata</code> if both are specified, with low precedence on <code>metadata</code> options.
debug	Leave intermediate files in place after render.
quiet	Suppress warning and other messages.
profile	Quarto project profile(s) to use. Either a character vector of profile names or NULL to use the default profile.
quarto_args	Character vector of other <code>quarto</code> CLI flag pass to the command. This is mainly for advanced usage, e.g it can be useful for new options added to <code>quarto</code> CLI and not yet supported as function argument.
pandoc_args	Additional command line options to pass to <code>pandoc</code> .
as_job	Render as an RStudio background job. Default is "auto", which will render individual documents normally and projects as background jobs. Use the <code>quarto.render_as_job</code> R option to control the default globally.

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")

# Override metadata
quarto_render("notebook.Rmd", override = list(lang = "fr", echo = "false"))

## End(Not run)
```

quarto_serve *Serve Interactive Document*

Description

Serve a Shiny interactive document. By default, the document will be rendered first and then served. If you have previously rendered the document, pass `render = FALSE` to skip rendering.

Usage

```
quarto_serve(
  input,
  render = TRUE,
  port = getOption("shiny.port"),
  host = getOption("shiny.host", "127.0.0.1"),
  browse = TRUE
)
```

Arguments

input	The input file to run. Should be a file with a <code>server: shiny</code> entry in its YAML front-matter.
render	Render the document before serving it.
port	Port to listen on (defaults to 4848)
host	Hostname to bind to (defaults to 127.0.0.1)
browse	Open a browser to preview the content. Defaults to using the RStudio Viewer when running within RStudio. Pass a function (e.g. <code>utils::browseURL</code>) to override this behavior.

quarto_use_template *Use a custom format extension template*

Description

Install and use a template for Quarto using `quarto use`.

Usage

```
quarto_use_template(
  template,
  no_prompt = FALSE,
  quiet = FALSE,
  quarto_args = NULL
)
```

Arguments

template	The template to install, either an archive or a GitHub repository as described in the documentation https://quarto.org/docs/extensions/formats.html .
no_prompt	Do not prompt to confirm approval to download external extension.
quiet	Suppress warnings and messages.
quarto_args	Character vector of other quarto CLI flag pass to the command. This is mainly for advanced usage, e.g it can be useful for new options added to quarto CLI and not yet supported as function argument.

Examples

```
## Not run:  
# Install a template and set up a draft document from a GitHub repository  
quarto_use_template("quarto-journals/jss")  
  
# Install a template and set up a draft document from a ZIP archive  
quarto_use_template("https://github.com/quarto-journals/jss/archive/refs/heads/main.zip")  
  
## End(Not run)
```

quarto_version	<i>Check quarto version</i>
----------------	-----------------------------

Description

Determine the specific version of quarto binary found by `quarto_path()`. If it returns 99.9.9 then it means you are using a dev version.

Usage

```
quarto_version()
```

Value

a `numeric_version` with the quarto version found

Index

accounts(), 9

deployments(), 9

is_using_quarto, 2

numeric_version, 13

quarto_add_extension, 3

quarto_binary_sitrep, 4

quarto_create_project, 4

quarto_inspect, 5

quarto_path, 6

quarto_path(), 13

quarto_preview, 7

quarto_preview_stop (quarto_preview), 7

quarto_publish_app
 (quarto_publish_doc), 8

quarto_publish_doc, 8

quarto_publish_site
 (quarto_publish_doc), 8

quarto_render, 10

quarto_serve, 12

quarto_use_template, 12

quarto_version, 13

quarto_version(), 5