Package 'quarto'

January 6, 2022

Title R Interface to 'Quarto' Markdown Publishing System

Version 1.1

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 2022-01-06 21:20:02 UTC

R topics documented:

quarto *Internal package state*

Description

Internal package state

Usage

```
quarto
```

Format

An object of class `environment` of length 0.

quarto_inspect *Inspect Quarto Input File or Project*

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 = ".")
```

Arguments

`input`	The input file or project directory to inspect.

Value

Named list. For input files, the list has members engine, format, and resources. For projects the list has members engines and config

Examples

```
## Not run:
# Inspect input file file
quarto_inspect("notebook.Rmd")

# Inspect project
quarto_inspect("myproject")

## 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)

quarto_preview *Quarto Preview*

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. utils::browseURL to override this behavior).
watch	Watch for changes and automatically reload browser.
navigate	Automatically navivate 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 RStudio Connect and ShinyApps

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

`input`	The input file or project directory to be published. Defaults to the current working directory.
`name`	Name for publishing (names must be unique within an account). Defaults to the name of the `input`.
`title`	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
`server`	Server name. Use "shinyapps.io" when deploying applications to Shinyapps. Use "rpubs.com" when deploying documents to RPubs. Otherwise use the domain name or IP address of any RStudio Connect server.
`account`	Account to deploy application to. This parameter is only required for the initial deployment of an application when there are multiple accounts configured on the system (see accounts).
`render`	`local` to render locally before publishing; `server` to render on the server; `none` to use whatever rendered content currently exists locally. (defaults to `local`)
`metadata`	Additional metadata fields to save with the deployment record. These fields will be returned on subsequent calls to `deployments()`.

...	Named parameters to pass along to `rsconnect::deployApp()`

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` *Render Markdown*

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,
  cache = NULL,
  cache_refresh = FALSE,
  debug = FALSE,
  quiet = FALSE,
  pandoc_args = NULL,
  as_job = getOption("quarto.render_as_job", "auto")
)
```

Arguments

input	The input file or project directory to be rendered (defualts 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.
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.
cache	Cache execution output (uses knitr cache and jupyter-cache respectively for Rmd and Jupyter input files).
cache_refresh	Force refresh of execution cache.
debug	Leave intermediate files in place after render.
quiet	Suppress warning and other messages.
pandoc_args	Additional command line options to pass to pandoc.
as_job	Render as an RStudio background job. Default is "auto", which will render individual documents normally and projects as background jobs. Use the quarto.render_as_job 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")

## 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 serve Should be a file with a `server: shiny` 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. `utils::browseURL` to override this behavior).

Index