

Package ‘qpost’

September 17, 2026

Title Create a 'Quarto' Blog Post

Version 1.1.0

Description Provides an interactive 'RStudio' dialog for creating 'Quarto' blog posts with correctly structured YAML front matter. The dialog collects title, author, date, categories, and other metadata, then scaffolds the post directory, creates the 'index.qmd' file, and optionally copies an image. A companion function appends COinS (ContextObjects in Spans) metadata to posts for automatic bibliographic import into reference managers such as 'Zotero'.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

URL <https://github.com/petzi53/qpost>,
<https://www.peter-baumgartner.net/qpost/>

BugReports <https://github.com/petzi53/qpost/issues>

Imports fs, glue, here, htmltools, lubridate, miniUI, purrr, readr,
rlang, rstudioapi, shiny, shinyFeedback, stringi, stringr,
urltools, yesno, yaml

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Peter Baumgartner [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0003-4526-8791>>)

Maintainer Peter Baumgartner <petzi53@gmail.com>

Repository CRAN

Date/Publication 2026-09-16 23:30:21 UTC

Contents

add_coins	2
edit_post	4
qpost	5
Index	6

add_coins	<i>Generate and append COinS metadata to a Quarto post</i>
-----------	--

Description

Automatically extracts bibliographic metadata from a Quarto post’s YAML front matter and injects it as a COinS (ContextObjects in Spans) metadata chunk. COinS enables bibliographic tools like Zotero to detect and import citation information from rendered HTML.

Usage

```
add_coins(file_path = NULL, backup = TRUE)
```

Arguments

file_path	Path to the .qmd file to process. If NULL (the default), the file is auto-detected as described in "Determining the Target File" below.
backup	Logical. Create a .bak backup before modifying the file (default TRUE). Recommended for peace of mind during development.

Details

Setup:

To use add_coins() effectively, configure your project’s .Rprofile (at the same level as _quarto.yml) with:

```
options(  
  qpost.lang = "en",  
  qpost.license = "CC BY 4.0"  
)
```

Then restart your R session. These settings provide defaults for the lang and license fields, which can be overridden per-post.

Auto-resolution of Metadata Fields:

Four YAML fields are automatically resolved in this priority order:

1. Value in the document’s YAML header (if present and non-empty)
2. Derived from _quarto.yml (blog-title and url only)
3. From .Rprofile options (lang and license only)
4. Omitted if not found anywhere

The required fields title and date must be present in the YAML header.

Generated Output:

The function appends a fenced R code cell (`{r}` block) with output rendered as a hidden `<span class="Z3988">` containing the COinS query string. This metadata is machine-readable but invisible in the browser. If a COinS chunk already exists (detected by the label `coins-code`), the user is prompted to confirm overwrite, and a `.bak` backup is created.

Determining the Target File:

`file_path` is resolved in this order:

1. The `file_path` argument, if supplied.
2. The active document in RStudio or Positron, via `rstudioapi`.
3. An interactive file picker (`file.choose()`) in other interactive R sessions (e.g. a plain R console).
4. Otherwise, the function stops with a message asking for `file_path` to be supplied directly.

Value

Invisibly returns the generated COinS chunk as a character string. Messages and the chunk code are printed to the console.

Examples

```
# Create a minimal Quarto project in a temporary directory
tmp <- tempfile()
dir.create(tmp)
writeLines(
  c("project:", " type: website", "website:", " title: My Blog",
    " site-url: https://example.com"),
  file.path(tmp, "_quarto.yml")
)
post_file <- file.path(tmp, "index.qmd")
writeLines(
  c("---", "title: Test Post", "date: 2026-01-01", "---", "", "Content."),
  post_file
)
add_coins(file_path = post_file, backup = FALSE)

## Not run:
# From an RStudio or Positron editor with a .qmd file open:
add_coins()

# To skip backup:
add_coins(backup = FALSE)

# To process a file directly (no IDE required):
add_coins(file_path = "path/to/post.qmd")

## End(Not run)
```

edit_post

*Edit an Existing Quarto Blog Post's YAML Header***Description**

`edit_post()` opens the same interactive dialog as `qpost()`, pre-populated with the metadata from an existing post. On submit it rewrites only the YAML front matter, leaving the post body untouched.

Usage

```
edit_post(file_path = NULL, backup = TRUE)
```

Arguments

<code>file_path</code>	Optional path to a .qmd file. When NULL (the default), <code>edit_post()</code> uses the file currently open in the RStudio or Positron editor.
<code>backup</code>	Logical. If TRUE (the default), a .bak copy of the original file is created before any changes are made.

Details

`edit_post()` requires a pane-capable IDE (RStudio or Positron) because it relies on **rstudioapi** to display the dialog. It cannot be used in a plain R console or non-interactive script.

Changing the title

When you modify the title in the dialog and click Done, you are asked whether to create a new post directory for the new title:

- **Yes** — copies all files from the old post directory into a new directory named after the new title's kebab-case slug. The original directory is renamed to `_old-slug.bak/` and its `index.qmd` has `draft: true` set. The new file is opened in the editor automatically.
- **No** — updates only the YAML `title:` field; the directory name stays unchanged.

Important: if the post has already been published, creating a new directory changes its URL. This will break existing links, bookmarks, and search-engine entries pointing to the old address.

The backup directory is protected from accidental publication by two independent mechanisms: the leading `_` in its name causes Quarto to skip it during rendering (consistent with Quarto's own `_freeze/` and `_site/` convention), and `draft: true` in the YAML acts as an additional human-readable signal. To recover the old title, rename the directory by removing the leading `_` and the `.bak` suffix.

The dialog shows a reminder in the title field as soon as it detects a change, so you are always aware before clicking Done.

`date-modified`

This field is automatically set to today's date on every save.

Value

Nothing. The side effect is an updated YAML front matter in the target `.qmd` file.

Examples

```
edit_post()
```

`qpost`

Create a New Quarto Blog Post

Description

`qpost()` opens an interactive dialog for entering a post's title and other metadata, then scaffolds the corresponding blog post file.

Usage

```
qpost()
```

Details

`qpost()` requires a pane-capable IDE (RStudio or Positron) because it relies on **rstudioapi** to display the dialog and open the resulting file. It cannot be used in a plain R console or non-interactive script.

Value

Nothing. What matters are the side effects:

- create a folder named with the date followed by the kebab-case post title
- create an `index.qmd` file
- copy the (optional) image file into the folder
- populate the YAML front matter
- open the new file in RStudio or Positron for editing

Examples

```
qpost()
```

Index

`add_coins`, [2](#)

`edit_post`, [4](#)

`qpost`, [5](#)

`qpost()`, [4](#)