Package: bd2q (via r-universe)

August 17, 2024

Title Convert a 'blogdown' Blog to Quarto

Version 0.0.0.9000

Description Takes the folder stucture and files of an old 'blogdown'
blog and translates them to what's needed for a Quarto blog.
Built for my own personal blog, so your mileage may vary.

URL https://github.com/matt-dray/bd2q

BugReports https://github.com/matt-dray/bd2q/issues

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Imports cli, fs, purrr, readr, stringr

Repository https://matt-dray.r-universe.dev

RemoteUrl https://github.com/matt-dray/bd2q

RemoteRef HEAD

RemoteSha b53d4bacaec14f5d8e4f88b1c3fd087c15df8908

Contents

create_and_transfer *Transfer a 'blogdown' Blog to Quarto Structure*

Description

A convenience function that creates a Quarto blog template and transfers the structure and content of posts and resources from a 'blogdown' blog project.

Usage

```
create_and_transfer(
  bd_path,
  q_path,
  rproj = TRUE,
  resources_dir = "resources",
  exts_keep = c("gif", "jpg", "jpeg", "png", "svg", "wav"),
  overwrite = FALSE
)
```

Arguments

`bd_path`	Character. Path to directory containing a 'blogdown' blog from where you want to copy posts
`q_path`	Character. Path to directory containing a Quarto blog where you want to copy posts to.
`rproj`	Logical. Include an R Project (.Rproj) file? Defaults to `TRUE`.
`resources_dir`	Character. The name of the folder that contains resources, such as images, that are used in posts.
`exts_keep`	Character. A vector of extensions for files to keep from the 'blogdown' blog.
`overwrite`	Logical. Overwrite the existing directory and all posts and resource files at the provided q_path? Defaults to `FALSE`.

Details

Executes create_template, transfer_posts and transfer_resources to set up the basic Quarto blog template and generate and fill the posts/ directory based on the posts found in the provided 'blogdown' blog project

Value

Nothing. New directory structure and files are created at the path given by q_path.

Examples

```
## Not run: create_and_transfer
```

`create_template` *Generate a Quarto Blog Template*

Description

Create template structure and files for a basic Quarto blog.

Usage

```
create_template(path, rproj = TRUE, overwrite = FALSE)
```

Arguments

`path`	Character. Path to a directory where you want to create a Quarto blog.
`rproj`	Logical. Include an R Project (.Rproj) file? Defaults to `TRUE`.
`overwrite`	Logical. Overwrite an existing directory at the provided path? Defaults to `FALSE`.

Details

Creates a blog directory at the user-specified path, with the files _quarto.yml, about.qmd, index.qmd, posts/_metadata.yml, styles.css and an optional .Rproj file. This is similar to the starter blog created in the Quarto quickstart guide, but with some opinionated differences.

Value

Nothing. New directory structure and files are created at the path provided by the user.

Examples

```
## Not run: create_template("~/Documents/new-quarto-project")
```

`remove_line` *Remove a Line from a Quarto Post*

Description

Search each Quarto blog post's index.qmd file for lines that contain a supplied regular expression, delete them and overwrite the original file.

Usage

```
remove_line(q_path, detect_rx)
```

Arguments

`q_path`	Character. Path to directory containing the Quarto blog.
`detect_rx`	Character. A regular expression that captures text within a line that you want to delete from each post's index.qmd file.

Value

Nothing. Quarto files are overwritten if they contain matching lines.

Examples

```
## Not run:
remove_line(
  q_path = "~/Documents/new-quarto-project/",
  detect_rx = "^draft:"  # remove YAML line that specifies draft status
)

## End(Not run)
```

`replace_lines` *Replace or Remove Lines from a Post*

Description

Replace or Remove Lines from a Post

Usage

```
replace_lines(q_path, match_str, replacement_str, collapse_str = "///")
```

Arguments

`q_path`	Character. Path to directory containing the Quarto blog.
`match_str`	Character. A vector of strings that match exactly to consecutive lines in a blog post that you want to replace (with 'replacement_str') or remove (when 'replacement_str' is `NULL`).
`replacement_str`	Character. A vector of consecutive strings to replace the strings proviede in 'match_str'. Specify `NULL` (default) if you want to remove 'match_str' without replacement.
`collapse_str`	Character. The function works by collapsing a post's lines to a single string, with elements separated by some unique string.

Value

Nothing. Quarto files are overwritten if they contain matching lines.

Examples

```
## Not run:
replace_lines(
  q_path = "~/Documents/new-quarto-project/",
  match_str = c("This is an example", "These are consecutive lines to match"),
  replacement_str = c("Replace with this line", "And this one")
)

## End(Not run)
```

---

transfer_posts *Transfer Posts from 'blogdown' to Quarto*

Description

Generate the folder structure required to move R Markdown (.Rmd) posts from a 'blogdown' project to a Quarto project and then copy them over, naming each one 'index.qmd' in its own folder.

Usage

```
transfer_posts(bd_path, q_path, ...)
```

Arguments

`bd_path`	Character. Path to directory containing a 'blogdown' blog from where you want to copy posts
`q_path`	Character. Path to directory containing a Quarto blog where you want to copy posts to.
`...`	Passed to fs::file_copy, with the intention that you can supply `overwrite = TRUE` if you need to overwrite any existing files.

Details

Assumes that Rmd posts in the 'blogdown' blog are at the path /content/post/YYYY-MM-DD-post-name.Rmd. For Quarto, the corresponding path will be /posts/YYYY-MM-DD-post-name/ and the file containing each post will be named index.qmd. (Each of these individual post directories can also hold a folder of associated resources for each post, which can be generated and copied from a 'blogdown' blog project using the transfer_resources function.)

Value

Nothing. New directory structure and files are created at the path given by q_path.

Examples

```
## Not run:
transfer_posts(
    bd_path = "~/Documents/old-blogdown-project/",
    q_path  = "~/Documents/new-quarto-project/",
    overwrite = TRUE
)

## End(Not run)
```

transfer_resources *Transfer Resources from 'blogdown' to Quarto*

Description

Copy resources (images, audio files, etc) from the directory structure of a 'blogdown' blog to that required by Quarto. Assumes you've already set up the directory structure for each post using transfer_posts.

Usage

```
transfer_resources(
  bd_path,
  q_path,
  resources_dir = "resources",
  exts_keep = c("gif", "jpg", "jpeg", "png", "svg", "wav"),
  ...
)
```

Arguments

`bd_path` Character. Path to directory containing a 'blogdown' blog from where you want to copy resources.

`q_path` Character. Path to directory containing a Quarto blog where you want to copy resources to.

`resources_dir` Character. The name of the folder that contains resources, such as images, that are used in posts.

`exts_keep` Character. A vector of extensions for files to keep from the 'blogdown' blog.

`...` Passed to fs::file_copy, with the intention that you can supply `overwrite = TRUE` if you need to overwrite any existing files.

Details

Assumes that the resources for a given 'blogdown' post are at the path /static/post/YYYY-MM-DD-post-name_files/, and the that the corresponding Quarto project will have a subdirectory for resources, which defaults to the name 'resources/', in the folder /posts/YYYY-MM-DD-post-name/ alongside that post's index.qmd file.

Value

Nothing. New directory structure and files are created at the path given by q_path, in the subdirectory given by resources_dir.

Examples

```
## Not run:
transfer_resources(
    bd_path = "~/Documents/old-blogdown-project/",
    q_path  = "~/Documents/new-quarto-project/",
    resources_dir = "resources",
    exts_keep = c("gif", "jpg", "jpeg", "png", "svg", "wav"),
    overwrite = TRUE
)

## End(Not run)
```

`update_resource_paths` *Update Resource Paths*

Description

Find lines from each post that contain paths to resources and update them from 'blogdown' to Quarto structure.

Usage

```
update_resource_paths(
  q_path,
  resources_dir = "resources",
  resource_rx = "<img src="
)
```

Arguments

`q_path` Character. Path to directory containing the Quarto blog.

`resources_dir` Character. The name of the folder that contains resources, such as images, that are used in posts.

`resource_rx` Character. A regular expression that will capture a line from a qmd file that contains resource paths to be edited.

Details

Assumes that the resources for a 'blogdown' post are at the path /static/post/YYYY-MM-DD-post-name_files/, while for Quarto there's a subdirectory for resources (defaults to 'resources/) in /posts/YYYY-MM-DD-post-name/ alongside that post's index.qmd file.

Value

Nothing. New directory structure and files are created at the path given by q_path, in the subdirectory given by resources_dir.

Examples

```
## Not run:
update_resource_paths(
  q_path = "~/Documents/new-quarto-project/",
  resources_dir = "resources",
  resource_rx = "<img src="  # lines that contain images
)

## End(Not run)
```

Index