Publish with Github Pages

Darren Irwin

2023-05-21

These are notes on how I set up a basic website to be published using Github Pages (and with benefit of version control using Github).

I assume you already are set up with Github and have that linked to VS Code. (If not, search for tutorials on that topic.) I am also assuming you have set up Quarto in VS Code–you can see my notes on that here .

One important thing to note: For individual users, Github Pages is only for publishing public websites. If you want a private (e.g. password-controlled) website, it is best to use some other hosting service.

I thank Joshua French for his excellent Youtube guide which helped me with some of these steps.

Create a Quarto website project

In VS Code, execute the "Quarto: Create Project" command from the command-palette. Choose "Website Project". Choose the directory on your computer within which the project directory will be created and enter your choice for a name for your website project directory.

The above will create the directory and a set of files for a basic website. The home page is `index.qmd` and there is also an "about" page in `about.qmd`. There is also a `_quarto.yml` file, which is a configuration file that provides website options and defaults for pages on the site. This can be opened and inspected in VS Code.

With the `index.qmd` file open, you can click "Render HTML" (at the upper right) and this should produce an HTML preview of the homepage in the window to the right.

Set up correct render directory

For working correctly with Github Pages, we need to render to a directory called `docs`. To do this, add `output-dir: docs` under the `project` heading in the `_quarto.yml` file. This heading section will then look like this:

```
project:
  type: website
  output-dir: docs
```

Save that and press render, and `docs` folder should appear in the list of files. You can now delete the `_site` folder.

Create Github repository

Login to your Github account using a web browser, and create a new repository. Give it the same name as your local project directory. You don't need to initialize with a readme, gitignore, or license.

Then choose "uploading an existing file".

Then drag all files in your local project directory (the files created above) into the window.

Add brief description (e.g., "initial basic website setup") and press Commit changes.

Click Settings (upper right), then Pages (along left).

Under Branch, select Main.

Change "/root" folder to "/docs", then press Save.

Wait a bit, then refresh page, and see near top that "Your site is live at . . ." You can click that link to visit your new webpage.

Set up link between local and Github directories

To set up to render and sync local folder to Github repo, you still need to link the full local folder to the Github repo. One way to do this is:

On the VS Code welcome page, choose "Clone Git Repository", then choose the name of the Github repo, then choose the local project directory, then click on "Select as Repository Destination".

Then, the local folder is the same as the Github one, and any changes can be Staged, Committed, and Synchronized (if this doesn't make sense, search out some tutorials on Git and Github, and working with VS Code in the context of those).

I don't know if this is necessary, but I added an empty `.nojekyll` to my project directory, as recommended here: https://quarto.org/docs/publishing/github-pages.html . This tells Github not to do any additional processing of the published site

An easy way to do the above: In Terminal window (at base of VS Code window), in your project directory: type `touch .nojekyll` which creates an empty file.