AsciiDoctor GitHub Pages Action

An always updated version of this document is available here as a PDF e-book.

A GitHub Action that builds AsciiDoc GitHub Pages in your CI workflow. It recursively converts every `adoc` file to `html`, renaming resulting `README.html` to `index.html` then pushing all generated html and existing files to the `gh-pages` branch. If you don't need anything fancy like Antora, this action might be the way to go to publish a simple AsciiDoc website. But if you need a more structured website, maybe Jekyll AsciiDoc QuickStart is for you.

After configuring the action, your GitHub Pages will be available at `http://your-username.github.io/your-repository`.

Keep in mind that every time the action is executed, the `gh-pages` branch is wiped out. If you manually add anything to it, outside of the CI workflow, the content will be lost.

1. Project on GitHub

View the project on GitHub and take the chance to give it a star.

2. Configuration

You have to just add the action to your yml workflow file and that is it. You can optionally customize the build by giving extra parameters to the action, which will be handed to the asciidoctor tool.

You can check a complete workflow file here. If you don't want to use the GitHub Action interface and just copy that file to the same place inside your repository, it may work out of the box.

2.1. Troubleshooting

If you get the error "remote: Permission to git denied to github-actions[bot]", access `Settings > Actions > General` and select `Read and write permissions`.

2.2. Building an e-book

The action allows enabling the automatic generation of an `ebook.pdf` file from the AsciiDoc files. The pdf is pushed to the `gh-pages` branch too. To enable that, just add the following configuration:

```
pdf_build: true
```

2.3. AsciiDoctor Reveal.js Slides

You can also build AsciiDoctor Reveal.js slides with this action. That will generate a slides.html file into the `gh-pages` branch. You can use the following configuration for that:

- `slides_build: boolean` - enables building a slides.html file (default false)
- `slides_main_adoc_file: string` - defines the name of the AsciiDoc source file to build the slides (default 'README'). **Do not include the file extension.**
- `slides_skip_asciidoctor_build: boolean` - to enable skippig the build of regular html files using the asciidoctor command, if you just want to generate the slides (default false)

3. Other examples

If you want to check how to create a website from multiple AsciiDoc documents, check this sample repository. It's only in Portuguese, but you can get the structure.

How the action works

The action is very simple. It's fired everytime commits are pushed to a branch or pull request (PR). Everything happens inside a container created on GitHub servers to execute the action. Then, the following steps are performed inside the container:

1. The pushed branch or PR is fetched in order to get the updated files in your repository.
2. Those files are copied to the gh-pages branch. The branch is created if it doesn't exist, or overriden otherwise.
3. Pre-build command (optionally provided in the `pre_build` parameter of your workflow) is executed, to perform any task you want before the AsciiDoc files are built.
4. Then, every AsciiDoc file is built to html and added to the gh-pages branch.
5. AsciiDoc files are removed from the gh-pages branch. All other files are kept, except yml configuration and .github/ directory.
6. Post-build command (optionally provided in the `post_build` parameter of your workflow) is executed, to perform any task you want after the AsciiDoc files are built.
7. Changes in the gh-pages are commited and pushed to you repository, publishing your updated website.

1. FAQ

1.1. How can I configure a custom domain?

GitHub Pages need a `CNAME` file on the `gh-pages` branch. But on every action run, the `gh-pages` branch is wiped out. To make a custom domain work, just add the `CNAME` file in the root directory of your `main` or `master` branch instead and it will be copied over to the `gh-pages` branch automatically. If you

have a `source_dir` configured, the `CNAME` file must be inside your configured `source_dir`.