Contribution Guide

See something on this site that is inaccurate, missing, or that could simply be improved? There are multiple ways for you to help make this site better, and we welcome all of them.

Technical Writing Resources

Whether you are an experienced technical writer, a regular (or not-so-regular) open source contributor, or you're making your very first writing contribution, there are a ton of helpful resources on technical writing. Here are a few:

- Digital Ocean's Technical Writing Guidelines

- Red Hat's Writing Style Guide

- Gareth Dwyer's Technical Writing repository

- Bolaji Ayodeji's Awesome Technical Writing collection

Markdown

The documentation on this site uses Markdown. Markdown is a documentation format that is rich enough to be useful for good technical documentation, and yet simpler and easier to learn than other formats like reStructuredText or DocBook XML.

If you're unfamiliar with Markdown, you can read up on its basics in this classic article by John Gruber if you're interested, but chances are that you'll also find all the information you'll need in this cheat sheet by Adam Pritchard, or the Start Writing guide from GitHub.

Or you simply look at the source of one of the pages on this site (try the Edit button on this one!) and figure it out as you go along — it's really pretty straightforward.

Modifications to content on this site

You have two options for editing content: directly in your browser using GitHub, or using a Git-based workflow from your local work environment.

Making contributions from your browser

Every page on this site has an Edit button ✏️. If you click it, it'll take you straight to the corresponding source page in GitHub. Then, you can follow GitHub's documentation on how to propose changes to another user's repository.

Making contributions using Git

The Git repository for this site lives at https://github.com/citynetwork/docs. You can fork that repository, make the proposed changes in your fork, and then send us a standard GitHub pull request.

If you would like to see your changes as you are working on them, you can do that with tox:

```
tox -e serve
```

A local copy of the documentation will then run on your local machine and be accessible from http://localhost:8000 in your browser.

Quality checks

There are a few checks that we apply to the configuration. They run automatically via GitHub Actions workflows when you send your PR:

- We check the commit message with gitlint, and enforce the Conventional Commits commit message style.

- We check whether the documentation still builds correctly, with your change applied.

- We check to make sure that no internal or external links in the documentation are dead. This is one example where the checks might fail through no fault of yours at all — some external link may have disappeared between the most recent change and your contribution, by pure coincidence. When that happens, we'll fix it together.

- We check some YAML conventions with yamllint. However, most contributions would probably only touch Markdown files and not YAML, so you're unlikely to trip over this.

If you're working in your local Git repository and your work environment has tox installed, you can also run the checks locally:

```
tox
```

You can also configure your local checkout to run quality checks on each commit. To do that, run:

```
git config core.hooksPath .githooks
```

License

With the sole exception of trademarks like "Cleura" and the Cleura logo, the content on this site is available under the Creative Commons Attribution-ShareAlike 4.0 International license, as you can see from the CC BY-SA icon at the bottom of each page. Please keep in mind that you are making your contribution under those terms.