Data Steward Circle

Contribution to DataPLANT

July 25th, 2023

Dominik Brilhaus

Knowledge Base

Create, share, reuse, collaborate on

- articles
- guides
- images
- slides
- ...

DataPLANT Knowledge Base: https://nfdi4plants.org/nfdi4plants.knowledgebase/

Content

- How to contribute to the DataPLANT Knowledge Base
- ... and DataPLANT projects in general

Resources

Contents shown here are (in part) already somewhere in the Knowledge Base or at least in the GitHub repo. However, ⚠️ Links there are somewhat volatile ⚠️

- Contribution Guide
- Markdown tutorial
- Marp slide decks

Part 1: Setup

Required

- ☑ a GitHub account
- ☑ Git installed and configured

Recommended software

☑ GitHub Desktop

- git synchronization
- pull requests
- manage (local and remote) branches

☑ Visual Studio Code

- all-in-one: text editor + file explorer + command line + git + ...
- code highlighting and linting
- many extensions available

Recommended VS code extensions for markdown

- Markdown all in one
  - Creating a TOC, use of shortcuts, creating a table, copy/pasting a table from excel
- markdownlint
  - markdown linting and style checking
  - Structuring and formatting
- Code Spell Checker
  - basic spell checker that works well with code and documents
- Markdown Shortcuts
- Markdown PDF
  - Converting a markdown to PDF
  - 💡 this does not follow any style / layout, e.g. linked in the YAML heading

Creating markdown slides in VS code: marp

Marp for VS Code

Part 2: Basics

Markdown

- general idea: WYSIWYG
- add YAML metadata

💡 See the Markdown tutorial and references therein

Markdown Slides (with Marp)

- add `marp: true` to YAML metadata

💡 See the Marp slide decks tutorial

GitHub

The DataPLANT GitHub organization (https://github.com/nfdi4plants) and repositories are the place where most of the magic (tool development) happens.

💡 Most repositories are **public** and open for (user) feedback

Issues

- Project-related task (lists)
- Communicate needs
- Example: Swate issues

💡 See also: https://docs.github.com/en/issues/tracking-your-work-with-issues/quickstart

Exercise: Open a GitHub issue

1. Navigate to the Knowledge Base or another DataPLANT repo
2. Raise an issue, e.g.
    i. raise a question
    ii. report a bug
    iii. request a feature

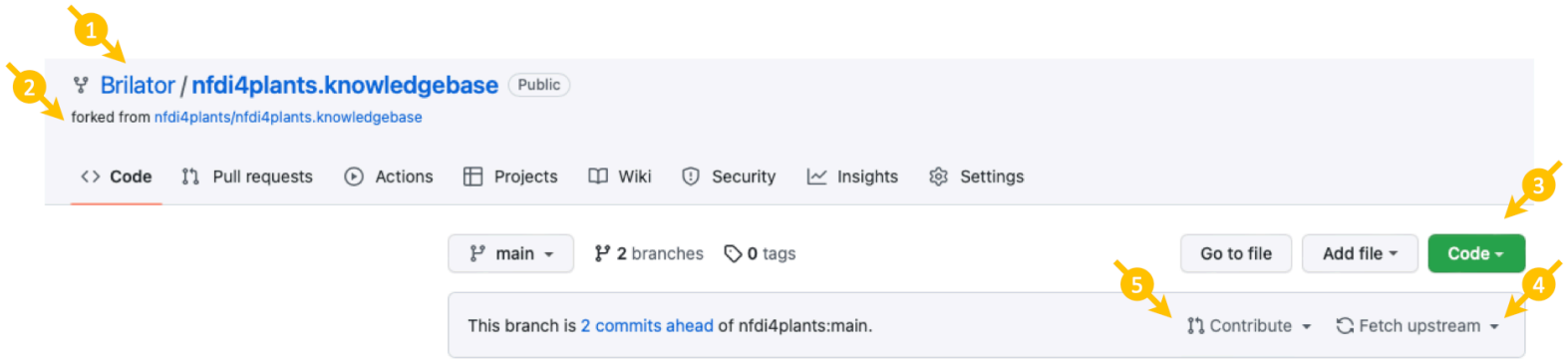
Forks and pull requests (PR)

- Fork = same project, different place
- Good routine to propose changes (contribute)
- Knowledge Base Forks

💡 See also: https://docs.github.com/en/get-started/quickstart/fork-a-repo

Exercise: Fork and clone a GitHub repo

1. Navigate to the Knowledge Base
2. Fork the repo (i.e. create a copy under your user name)
3. Clone your fork locally (to your machine)



👩‍💻 Exercise: Create a new article (markdown)

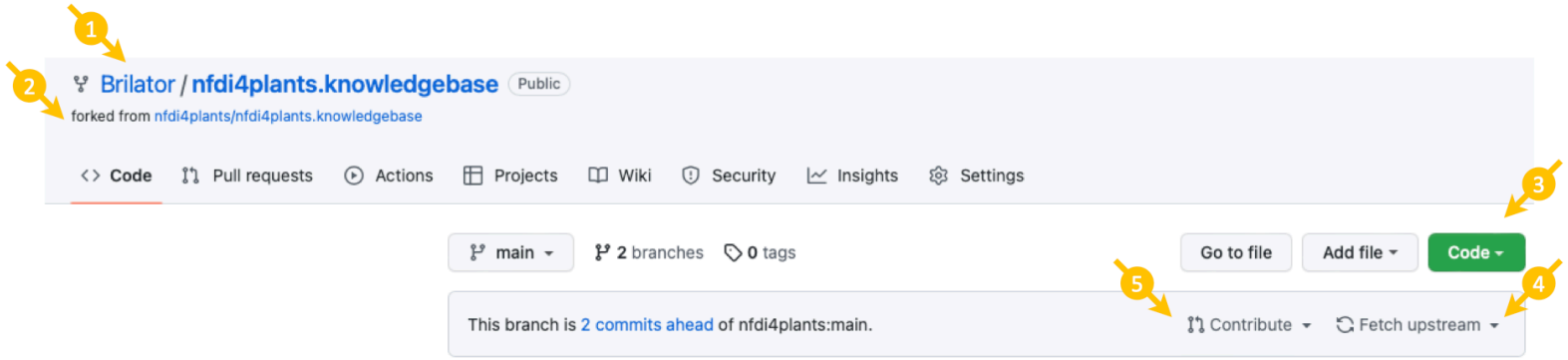
1. Open your local "Knowledge Base" clone (File explorer or VS Code)
2. In `src/docs/_ignored`, create a new text file `<YourInitials>_dsc.md`
3. Add the following YAML header:

```
---
layout: docs
title: <YourTitle>
author: <YourName>
date: 2023-07-25
---
```

4. add some text below the YAML header

Exercise: Open a pull request

1. Save your local changes
2. Sync to your fork
    i. via command line (add, commit, push) ***or***
    ii. via VS Code ***or***
    iii. via GitHub Desktop
3. Contribute / open a PR (follow instructions on GitHub)

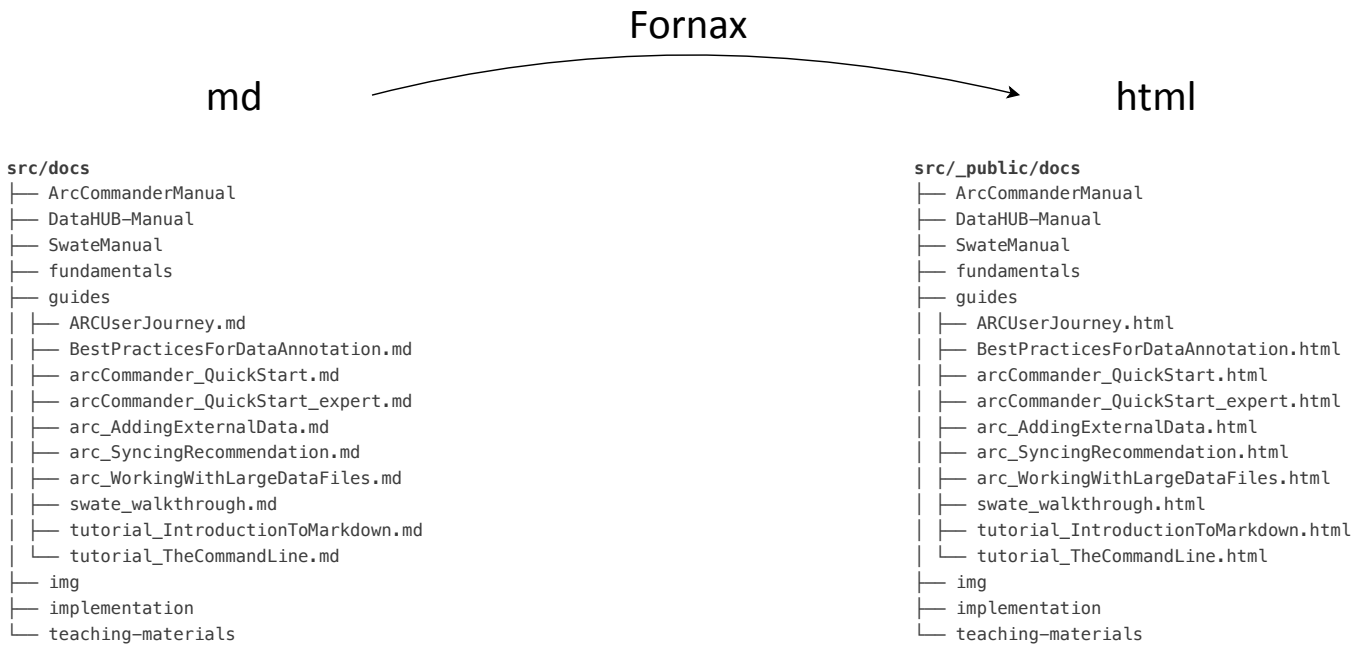


Part 3: The Knowledge Base

Required software

- ☑ Node.JS
- ☑ .NET
  - 💡 Both required for local testing of Knowledge Base

Knowledge Base backend: Fornax



💡 See nfdi4plants web components docs

Setup

1. Clone the repo.
2. Run `dotnet tool restore` in root directory.
3. Run `dotnet paket install` in root directory.
4. Run `npm install` in root directory.

💡 This needs to be done only once after cloning the repo.

Knowledge base local test

1. Run `npm run fornax`.
2. Open the page in your browser http://127.0.0.1:8080/.

Let's have a look at your articles (pull-requests)

Markdown syntax

- **MUST** start and end with `---` .
- **MUST** contain `layout: docs` .
  - This triggers fornax parsing to html.
- **MUST** contain `title: xxxx` .
  - added as "# xxxx" to the html
  - used to name the generated webpage.
- **MUST** contain `date: yyyy-MM-dd`
- MAY contain author: xxxx.
- MAY contain `add toc: true` .
  - If true, adds automated table of contents from all found headers in content.
- MAY contain `add support: true` .
  - If true adds DataPlant support component at the bottom.
- MAY contain `add sidebar: relative/path/to/sidebar.md` to add the sidebar element to the page.
- MAY contain any other metadata. The information will be read but will not affect the generated html.
- Keys (`layout` , `author` , etc.) are NOT case sensitive.
- Fields MAY be in any order.

```
---
layout: docs
title: Metadata
date: 2022-05-09
author: FirstName LastName
add toc: true
add sidebar: _sidebars/mainSidebar.md
---
```

💡 For details, check

- contribution guide and
- nfdi4plants web components docs

Slide decks – work in progress 🚧

- Fornax currently does not auto-convert `marp.md` to `marp.html`
- Slides are mostly stored in `src/docs/teaching-materials`
  - based on the concept presented before
- Naming convention and YAML header are up for discussion

💡 Feel free to create slide decks how and where you need them!

Now it's your turn

- Read and use the knowledge base
- Raise issues
- Open pull-requests to
  - suggest changes
  - create or edit articles, guides, tutorials