

2021-02-15 CTWG Meeting Notes

Meeting Date

- 15 Feb 2021

Attendees

- [Drummond Reed](#)
- [Rieks Joosten](#)
- [Daniel Hardman](#)
- [Scott Whitmire](#)
- [Brian Dill](#)
- Maria Iliadi (GRNET)

Main Goal of this Meeting:

To review the proposed new Docusaurus Terminology tooling and take one term through the entire lifecycle.

Agenda

Time	Item	Lead	Notes
5 min	Welcome & Antitrust Policy Notice Introduction of new members Agenda review	Chairs	
5 mins	Report on last meeting and updates on subsequent conversations	All	
40 mins	Review of Docusaurus Terminology tooling and end-to-end example of the lifecycle of a term	Daniel Hardman Maria Iliadi (GRNET)	
5 mins	Next steps	All	
2 mins	Review of Decisions and Action Items	Chairs	
1 min	Next meeting	Chairs	

Recording

- [Link](#)

Presentation(s)

- *None*

Documents

- [Docusaurus Terminology GitLab master](#)

Notes

1. New members
 - a. Maria is a software engineer who has been working at GRNET for over a year working mostly on Python and Javascript.
 - i. She has been working on Docusaurus Terminology in order to add terms to the documentation for packages using it.
2. Report on last meeting and updates on subsequent conversations
 - a. Rieks shared that he has shared in our Slack channel a summary of requirements/suggestions from eSSIF-Lab for Terminology Engine v2 (TEv2), where TEv1 is the current Docusaurus Terminology tool of GRNET.
3. Review of Docusaurus Terminology tooling and end-to-end example of a term lifecycle
 - a. Maria began by offering to do a walk-through of the Docusaurus Terminology capabilities.
 - b. The GRNET tool repo is here: <https://gitlab.grnet.gr/terminology/docusaurus-terminology/-/tree/master>
 - c. Maria showed us a number of capabilities of the current version of Docusaurus Terminology.
 - d. Daniel then asked Rieks to summarize what the delta is between the current version and what CTWG needs.
 - i. Rieks said that the primary delta is the need to support terms from multiple terminology owners or scopes.
 - ii. The second question Rieks had was around the use of double percent signs for delimiting terms.
 1. Maria noted that there has already been feedback about the inconvenience of using double percent signs.
 2. The Docusaurus Terminology team is already working on a more convenient method.
 3. This method would use the standard Markdown syntax of square brackets for a term followed by the term path in parentheses.
 - e. Rieks summarized that the Docusaurus Terminology tool works **primarily between the curation and the production**.
 - f. Daniel shared that his overall reaction that there are several aspects of how this tool works

- i. It combines the management of raw terminology with the management of those terms within a particular artifact (e.g., a white paper).
 - ii. Maria clarified that you can generate a glossary without modifying an artifact.
 - g. Daniel then asked more questions about using Docusaurus Terminology with multiple repositories with different owners.
 - i. Rieks clarified that Docusaurus Terminology V1 creates a list of the glossary entries from a group of artifacts.
 - ii. The glossary itself becomes just another document, in which each term used in the definition of another term would itself be a popup and a link.
 - h. We discussed what we would need to do an end-to-end lifecycle test.
 - i. Pick one white paper.
 - ii. Make sure it is in the Markdown format needed.
 - iii. Run it through the Docusaurus Terminology tooling.
 - iv. See how well it develops the output we need.
 - i. Maria offered to help with the documentation.
- 4. Next steps
 - a. Daniel will set up the CTWG Sandbox repo.
 - b. Maria will check the steps for GitHub vs. GitLab and update documentation accordingly.
 - c. Rieks will provide his SSI governance white paper in the Markdown with the terminology marked in the syntax required by Docusaurus Terminology.
 - d. Daniel will hook up the CI/CD pipeline so that it automatically generates a new website based on the output.
 - e. Our goal is to demonstrate all of this working at our next meeting in two weeks.
 - f. Once that is working, Rieks suggested we can show how other ToIP authors can use these same tools.
- 5. Review of Decisions and Action Items
- 6. Next meeting

Slides

From Maria's presentation:

Docusaurus Terminology

docusaurus-terminology is a yarn package for creating a terminology structure in your [Docusaurus](#) project. This plugin allows you to use terms in your pages that 'stick out' of the surrounding text, while hovering over them makes a popup appear with a short explanation of the term and clicking on the term navigates the user to the page that documents the concept.

[My Site](#) [Docs](#) [Blog](#)

Host

A host is a person responsible for guests at an event or for providing **hospitality** during it.

The noun host refers to a person who receives and entertains guests at an event, providing **hospitality** during it.

B The friendly and generous reception and entertainment of guests, visitors, or strangers. **y**, a host is an animal, plant or person that provides a home for another organism — like a parasite.

Host also functions as a noun, meaning a multitude, horde, or great number. As in, "You will pick up a host of funky germs from drinking this water."

Generating the Terminology Documentation

When you are finished referencing terms and have written corresponding term pages, you can test this locally by running:

```
yarn docusaurus parse
```

This will replace all `%%term_text|term_name%%` occurrences with the React component supporting the required functionality.

Generating the Glossary Page

If everything works well with the above procedure, you can then generate a glossary page, by running:

```
yarn docusaurus glossary
```

This will generate a file in `./docs/glossary.md` where every term that has been mentioned above will be populated in the `glossary.md` page.

When to Generate the Terminology Docs

As the terminology plugin actually edits all markdown files, your Git repository will show changes in the `git diff` command. It is highly recommended to avoid committing the changes, as the plugin will no longer be able to detect patterns that have been altered.

Your best case scenario will be to use the scripts in CI, just before building and deploying the documentation.

The following example of a Gitlab CI job shows how to perform these steps in the CI environment:

```
...
generate-docs:
  image: node:lts
  stage: build
  before_script:
```

```
index.js 2.29 KB
1  const path = require("path");
2
3  const parser = require("./commands/parser.js");
4  const glossary = require("./commands/glossary.js");
5  const validateOptions = require("./validator.js");
6
7  const DEFAULT_OPTIONS = {
8    docsDir: "./docs/",
9    termsDir: "./docs/terms/",
10   termsUrl: "/docs/terms",
11   glossaryFilepath: "./docs/glossary.md",
12   patternSeparator: "¶",
13   noParseFiles: [],
14   noGlossaryFiles: [],
15   dryRun: false,
16   debug: false
17 };
18
```

Decisions

- We will proceed with an end-to-end demonstration of how glossary development will work with a single white paper contributed by Rieks.

Action Items

- [Daniel Hardman](#) will set up the CTWG Sandbox repo.
- [Maria](#) will check the steps for GitHub vs. GitLab and update documentation accordingly.
- [Rieks Joosten](#) will provide his SSI governance white paper in the Markdown with the terminology marked in the syntax required by Docusaurus Terminology.
- [Daniel Hardman](#) will hook up the CI/CD pipeline so that it automatically generates a new website based on the output.