

aboutcode.org

Technical Writing Project Proposal GSoD'21

PROPOSAL TITLE

Create and update ScanCode.io HowTo Guides and Tutorials

About our Organization

AboutCode.org is a community of developers who focus on Software Composition Analysis (SCA) tools (command line tools, web-based and API servers and applications) and data for identifying and tracking software origin, licensing and security vulnerabilities. SCA tools and data are essential to enable everyone to safely produce and use free and open source software. And we think that these essential SCA tools should themselves be FOSS!

AboutCode.org was started by nexB Inc. in 2013. We have many contributors from a growing FOSS community including students who have kept working with us after GSoC program participation and with contributions sponsored by nexB. As with many open source projects, we only know the identity of a subset of our users, but we know that our AboutCode software is used by several open source Foundations such as Eclipse, OW2, the FSFE and many projects such as ORT, REUSE and Tern and companies.

About our project

Project's problem

ScanCode.io is an open platform to script and automate Software Composition Analysis for many different use cases including development codebases, package dependencies, and Docker, Virtual Machine images, or other containers. With the ScanPipe feature of ScanCode.io, you can create any number of Pipelines to integrate SCA into your development process. A Pipeline may include libraries beyond those

available from AboutCode.org. This flexibility means that we need much more extensive HowTo, Tutorials, and Reference documentation to help our users build their own Pipelines. ScanCode.io also offers a Web UI that needs a new Tutorial and HowTo documentation.

Although we already offer a few Pipeline templates, our users need to be self-sufficient to adapt these templates or add new Pipelines in order to fit SCA into their own environment. We also want to encourage users to contribute back improvements to our templates or new Pipeline templates. Both of these objectives require more and better documentation.

Project's scope

Our focus for GSoD 2021 is Scancode.io and these specific tasks:

- Create a HowTo guide for integrating third-party libraries into a ScanCode.io Pipeline.
- Add a tutorial for adding a new Pipeline with a third-party library.
- Extend the HowTo Guides to cover Software Composition Analysis workflows based on ScanCode.io
- Upgrade the ScanCode.io Web UI documentation
- Create an introductory video [`<https://opensource.com/article/21/3/video-open-source-tools>`](https://opensource.com/article/21/3/video-open-source-tools) to show how the web UI is used.
- Update and improve the existing Pipe libraries reference API documentation (which is generated from code documentation "docstrings").
- Sync the new documentation set with the code to support continuous integration with code changes.

Several maintainers and contributors have committed to supporting the GSoD project. We have identified volunteers and interest for AboutCode.org, but we have not yet identified a specific technical writer for the GSoD project.

Measuring our GSoD project's success

Our primary success metric will be the reduction in GitHub Issues related to installation, usage or customization of ScanCode.io. We will add a label in GitHub to track Issues related to documentation. We expect to reduce the percentage of Issues related to documentation by 40% to 50% over the year following completion of the project.

During the course of the project, we will also ask users to participate in a survey about the completeness and quality of our new project documentation. We will create a numerical score from the survey answers.

We expect a net improvement of 30% in positive scores between the start of the project and its completion.

Budget

Our budget for the project is \$15,000 which will be allocated completely to the technical writer working on the project. We expect that the technical writer will work on our project $\frac{1}{3}$ to $\frac{1}{2}$ time over a period of 5 months (May to October).

Additional Information

Previous experience with technical writers or documentation:

We believe that documentation should be created, managed, and tested like code. With this in mind, we expect to include any technical writer directly into the corresponding project development team. This approach worked well for our former GSoD project participation in 2019 and we have adapted it for other contributors to our project documentation.

Our documentation builds are tested in CI/CD, along with linters and link checkers. Most of this documentation infrastructure was implemented based on work from our 2019 GSoD program.

We expect to build on the process and tools we have in place for documentation to quickly onboard a new technical writer and let them focus on new content structure, design, and creation.

Our mentors have significant experience working with technical writers from prior product development work at commercial software companies and our prior GSoD participation. We have not yet worked with professional technical writers for AboutCode.org projects.

Previous participation in Season of Docs, Google Summer of Code:

All of our mentors have participated in one or more Google Summer of Code programs since 2015. Two of them also have experience in mentoring Technical Writers in the Google Season of Docs programs for 2019.

One of our mentors participated as a student in Google Season of Docs 2019 (Report at https://aboutcode.readthedocs.io/en/latest/gsoD/gsoD_2019_report.html). For GSoD 2019, our project focused on documentation for our Scancode-Toolkit project. The first step was to move existing documentation from a GitHub wiki to ReadTheDocs (with Sphinx and other tools) in order to better link documentation with code as part of our overall CI process. The project then focused on adding tutorials and improving how-to and (command-line) reference documentation.

Based on experience from our 2019 GSoD project, we were able to confirm that RTD/Sphinx tools were a good fit for our projects and we have since moved the primary documentation for our other projects to RTD. We also piloted our use of the documentation framework of Tutorials, How-To Guides, Reference and Discussion (from Daniele Procida of Divio/Django) which we are applying to all of our projects as we improve their documentation.