Revamping and restructuring the Talawa Docs

Organization or Project: Talawa
(*https://palisadoesfoundation.github.io/talawa-docs/docs/*)

Organization Description: *Talawa was created to help community based organizations collaborate with their membership. These organizations include religious groups, non-profit charities, social groups and in limited cases, businesses*.

Authors: Dominic Mills-Howell, Michael Lue, Tasneem Koushar

Problem Statement

What problem were you trying to solve with new or improved documentation?

The majority of the code is undocumented or badly documented and was not growing in an organic manner with our ever increasing codebase. Our goal is to ameliorate this problem through the creation/refinement of constructing documentation for our repositories which is extensive enough to account for the latest features and developments, such that we are in a good position for further development with the pipelines already in place. Furthermore, we need to make it straightforward for first time contributors to become productive with our repositories.

Proposal Abstract

A brief summary of your original organization proposal. Link to the proposal page on your project site, if possible.

In order to reduce the learning curve of contributors and sysadmins alike we need to document our three main repositories to make them easier to use. We leverage CI/CD pipelines that automate the documentation process through comments in newly added/modified files which are parsed as HTML/Markdown files and then sent to the documentation repository. We have four objectives for our documentation (albeit only the first two were completed during the Season of Docs programme): Adding base documentation for each repository (Talawa, Talawa-API and Talawa-Admin); the creation of how-to guides and tutorials for getting started; the creation of visualizations

for various GraphQL schema and queries; Restructuring the current documentation page.

Link to Talawa's Season of Docs proposal page:
https://palisadoesfoundation.github.io/talawa-docs/docs/internships/gsod/gsod-ideas

Project Description

Creating the proposal

How did you come up with your Season of Docs proposal? What process did your organization use to decide on an idea? How did you solicit and incorporate feedback?

We had a slack channel filled with past participants and people who weren't successful when they applied for the Google Summer of Code programme under Talawa in the previous year, and we made good use of the channel by asking them to list their pain points when trying to become productive with the project. We looked at what was causing the most friction (creating the *friction log*) for new users when trying to get setup and become industrious for the three Talawa repos. Afterwards we made a list ranking the most important aspects to address based on the frequency of common issues presented. The most frequent issues were at the top of the list and the less frequent under them. There was no tie breaking rule.

Budget

Include a short section on your budget. How did you estimate the work? Were there any unexpected expenses? Did you end up spending less than the grant award? Did you allocate funds properly or were some items you budgeted for more/less/unnecessary? Did you have other funds outside of Season of Docs that you were able to use?

From the provided stipend of $13,514 USD, a sum of $10,000 USD was budgeted for two technical writers and $2,000 for 4 volunteers where the remaining sum is dedicated towards swag, processing fees as well as the necessary graphic design for documentation. We estimated the work from the precedent set by organizations/repositories that had similar sized codebases and also via weekly meetings amongst volunteers to plan and gauge what is required to be done as well as coordinate alongside the participants from the ongoing Google Summer of Code programme to avoid conflicts with the development of new features.

Throughout the entire course for the Season of Docs programme, the allocation of funds were appropriately allotted to cover the relevant expenses in their respective

areas that we have incurred, with the exception of graphic designer, where we instead used though funds to defray the costs for thanking the many contributors who provided valuable insights in the friction log, we sent the Talawa associated paraphernalia, such as t-shirts, mugs, etc. Fortunately, there were no unexpected expenses and we ended up spending the approximate amount.

No other funds outside of Season of Docs were used.

Participants

Who worked on this project (use usernames if requested by participants)? How did you find and hire your technical writer? How did you find other volunteers or paid participants? What roles did they have? Did anyone drop out? What did you learn about recruiting, communication, and project management?

The two participants selected for this project were Eva Sharma and Vedant Jain. The process for determining our participants was split into two aspects. The first aspect consisted of ranking their previous experience with technical writing. And the latter part was to see how well they could interact and understand the codebase. This consisted of presenting three writing samples from any or all of the three Talawa repositories. We then created an excel spreadsheet and ranked all the prospective candidates with a weighting of 40% for their previous experience and 60% for their writing samples.

The mentors for the Season of Docs programme were chosen based on their experience as mentors in Google Summer of Code. A notable exception was Tasneem who joined us as a mentor when she came across our organization and noticed it was similar to her Google Summer of Code project from the year prior. We were able to make the most of her experience with using automated pipelines for the creation of new documentation.

Recruiting top talent is hard work and this programme has taught us that you need to choose an approach that is flexible in allowing your prospective candidates to show their talents. If we had chosen an approach where we based our assessment solely on past experience we would have missed out on what our chosen participants were truly capable of. Communication is key in any undertaking of this nature and we can say that there was a good flow between participants and mentors during this programme that made the process streamlined and seamless. Bad communication could have set us back months since many aspects of this project were contingent on interfacing with the GSoC participants and making compromises. In this respect, our mentors benefited from having been software engineers that have led teams previously in their work life and understood the basic tenets of project management.

Timeline

Give a short overview of the timeline of your project (indicate estimated end date or intermediate milestones if project is ongoing).Did the original timeline need adjustment?

Since the beginning of the organization, we did not prioritize documentation, as a result our code repositories lacked the relevant notation to provide participants with context on how and where to navigate through the code.

The timeline was mapped out based on which repositories required the necessary documentation to help participants get started as well as avoiding conflicts with the participants from Google Summer of Code occurring in parallel.

The timelines can be observed from our Google Sheets below:

https://docs.google.com/spreadsheets/d/1P0uv_yNA9DEGfGsSX9OsL1eHKh1Zv9T3SF2jZtLrsb0/edit#gid=651823730

Results

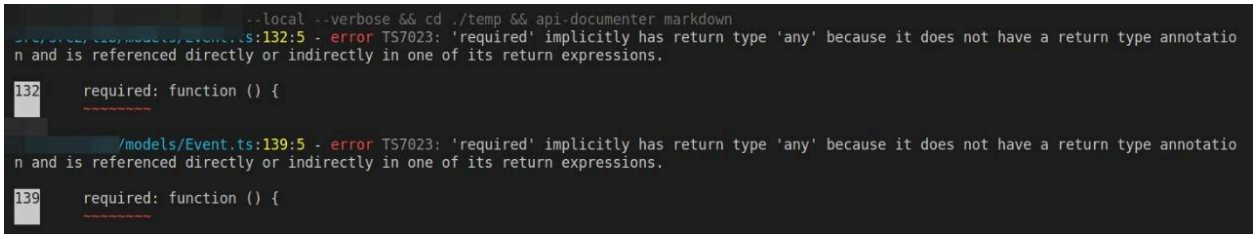
What was created, updated, or otherwise changed? Include links to published documentation if available. Were there any deliverables in the proposal that did not get created? List those as well. Did this project result in any new or updated processes or procedures in your organization?

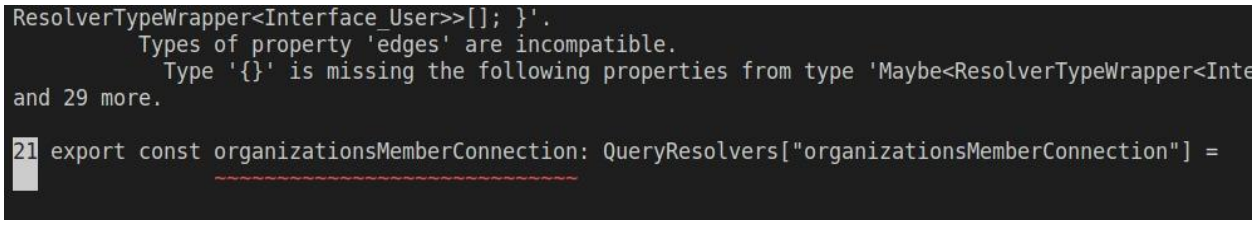
The process of documenting the three repositories was done through the use of CI/CD pipelines that used doc commenting tools such as jsdoc, dartdoc and SpectaQL to convert comments and code in any changed files, that were added to the repository, into HTML and/or Markdown artifacts and then sent to the documentation website repository where it would be reflected on the documentation website in real time. The workflow for the Talawa-API can be seen here (similar pipelines are in place for the frontend and admin panel repository):
https://github.com/PalisadoesFoundation/talawa-api/actions/workflows/pull-request.yml

The documentation process essentially details a modular approach towards the codebase by describing functions in terms of their respective inputs and outputs as well as describing how functions within designated files relate to other functions within and out of the respective files.

At the end of the Season of Docs programme, we had 100% of the Talawa-API code base documented, 70% of the Talawa-Flutter codebase documented and 40% of the Talawa Admin documented.

The (temporary) website hosting the documentation (shown in the image above) can be found at the link: https://ubiquitous-cranachan-212d00.netlify.app/. The documentation website (https://palisadoesfoundation.github.io) has not been updated due to the fact that pending merge conflicts that “break” the pipeline, from a typescript migration task which was a part of this year’s Google Summer of Code (GSoC) programme, need to be resolved. The issue stems from the fact that jsdoc was the primary tool for generating javascript documentation for the Talawa API repository and nearing the end of the GSoC programme. However, many of the Javascript files were to be rewritten as Typescript files as a task for one of our GSoC participants and the corresponding doc commenting tool tsdoc is far more rigid than its jsdoc counterpart and generated a number of errors related to the more rigorous type system imposed, examples of the errors encountered when parsing the typescript files to HTML/Markdown are shown below.





The outstanding deliverables are:

1. Resolving the merge conflicts and publishing the documentation to the dedicated hosted documentation site.
2. Refactoring and making the website more aesthetically pleasing. Much of the time spent during the programme was solely concentrated on getting the documentation in place. Getting the documentation in a more presentable manner will be completed by the mentors in the coming weeks.
3. Adding diagrams and various visualizations to better understand the complex processes and relationships with the repositories.
4. Fix grammar, spelling errors and awkwardly phrased sentences that may exist with the codebase (English was not the first language of either of the participants)

Lastly, since doc commenting libraries exist in many languages, such as Python, Java, etc. thus one can see how this approach could be refined and generalized for other use-cases.

Metrics

What metrics did you choose to measure the success of the project? Were you able to collect those metrics? Did the metrics correlate well or poorly with the behaviors or outcomes you wanted for the project? Did your metrics change since your proposal? Did you add or remove any metrics? How often do you intend to collect metrics going forward?

The rubric used for determining the success of the overall project was similar to that of the software testing metric used for testing certain blocks of code in a given file, more

commonly known as *code coverage*. As described in the previous response, the documentation was generated from comments in the files pushed to a repository that were converted to HTML/Markdown and then added to the documentation site. Checks were done in the pipeline to determine if the comments were of a particular form that matched those necessitated by the respective doc commenting tools also mentioned in the previous response. If a file had no comments then there was reasoned that there was no documentation for that particular file. Moreover, these checks gave us a heuristic to determine how much of the codebase was being documented. This is done as follows: a parser was created that counts the number of functions in a given file and determines the number of comments above a said function. If there exists a certain threshold of commented lines (the threshold chosen was three lines of comments) then a respective file was said to be fully documented if each function has at least three lines of comments about it in the format that a doc commenting library can parse.

As the codebase favors a more functional approach this augured well as a metric for determining how much of the codebase was documented overall which is reflected in the percentages given in the previous section.

The metric remained consistent throughout the proposal and the duration of the GSoD timeline. Nothing was removed and only further checks were added, such as parsers to determine if the structure of the comments were correct and that they were in the correct place.

Given the automated nature of the project we expect to further refine the process going forward and extrapolate any useful insights that can be gleaned from the Talawa project and adapted to other projects. We will be continuously collecting metrics for our repository as long as we have contributors to it.

Analysis

What went well? What was unexpected? What hurdles or setbacks did you face? Do you consider your project successful? Why or why not? (If it's too early to tell, explain when you expect to be able to judge the success of your project.)

The two selected participants, Eva and Vedent, were proactive in their duties and were able to become productive early on in the programme. They asked the relevant questions, leveraged the collective past experience of the slack group to better understand the intricacies of the project and communicated with their mentors whenever any issues arose.

What was unexpected and also a setback in a sense was the keeping apace with the documentation whilst the Google Summer of Code programme was running concurrently. There were periods where we had to have meetings with the GSoC participants to figure out the best route to proceed and develop a roadmap to guide the logistics of the GSoD and GSoC workflows. In many ways, this was an achievement in overcoming this obstacle as much of the current documentation reflects the most recent GSoC participants' changes, but at the same time it took much deliberation with, at times, many parties to determine how to proceed.

Overall, we consider the project to be a success as the Talawa-API is 100% documented and just needs polishing at this point. Whilst the other two repositories weren't documented to completion the work done in this programme provided an excellent starting point for what one might consider to be a deeply onerous task.

Summary

In 2-4 paragraphs, summarize your project experience. Highlight what you learned, and what you would choose to do differently in the future. What advice would you give to other projects trying to solve a similar problem with documentation?

All in all, the Season of Docs programme proved to be an enriching experience for Talawa for both the mentors and participants. Over the course of the programme, we developed the skills to manage different, and sometimes conflicting, groups, whilst managing time and meeting the demands of the project. In the initial stages, it was stressful to say the least and took a lot of consolidation from both the GSoC and GSoD groups to get the project to a stable place.

Had we had the chance to do it all over again, we would create a combined messaging group for both GSoC and GSoD participants (using Slack, Whatsapp, etc) where the participants could post the issues they were going to work on so that any conflicts could be resolved in real time. A common issue was that some tasks ballooned due to the branching strategy used (which was to create a develop branch and that would be then branched off into sub-issues which would be merged into the develop branch). Many of these issues could have been better dealt with more effective communication and if we had not tried to silo the GSoC and GSoD participants from one another.

My advice to other projects would be to communicate with the relevant bodies involved when creating documentation. In other words, look at who your contributors are and ask them of how they plan to contribute in the span of time the programme will be going on.

This will enable you to plan better and resolve seen and unforeseen conflicts in a more timely manner.

Appendix

If you have other materials you'd like to link to (for example, if you created a contract for working with your technical writer that you'd like to share, or templates for your documentation project, or other open documentation resources, you can list and link them here). The Appendix is also a good place to list links to any documentation tools or resources you used, or a place to add thanks or acknowledgments that might not fit into the sections above.