Have a complete OpenAPI compliant documentation for the Open Food Facts API

Organization or Project: Open Food Facts
https://wiki.openfoodfacts.org/SeasonOfDocs

Open Food Facts is the "wikipedia of food".

Make better food choices for your health and for the planet

Open Food Facts is a database of food and cosmetic products with all the data you can find on product labels.

Food additives, allergens, packaging codes: Open Food Facts helps you make sense of the fine print on products labels. Also, you can easily compare products in 3-clicks, so that you can make more informed choices.

Made by everyone

Open Food Facts is a non-profit association of volunteers. Since 2012, 25000+ contributors have added 2 Million+ products from 180 countries using the Android or iPhone apps to scan barcodes and upload pictures of products and their labels. We helped foster the Nutri-Score across Europe which is now present on physical projects, and we're doing the same for the planet with Eco-Score.

For everyone

Data about food and cosmetics is of public interest and has to be open. In addition to the official app, Open Food Facts is a platform for developers, and there are more than 150 re-uses of the data in many different ways: many nutrition apps to eat better, food inventory apps to prevent waste, research by health and nutrition scientists, investigations by journalists, educational games etc.

With a good documentation

As a result, many users (developers, scientists, innovators…) use and rely heavily on the Open Food Facts API. The better documentation we have, the more reuses we'll get, and the more apps we'll get to contribute back data to this common good.

Problem Statement

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

Many people rely on the Open Food Facts API to help transform the food system. Ensuring that the API documentation is OpenAPI compliant - but also clear and readable - will promote the development of new applications based on the Open Food Facts database, scientific reuses, and improve the quality and quantity of data. This, in turn, will increase the number of conscious customers checking via apps the nutrition values of the food products they purchase and support Open Food Facts' vision of making nutritional information available to everyone.

As a result, we made lowering the API barrier to entry and developing API data contributions two of our 2022 priorities. This project will thus help us reach those goals.

Proposal Abstract

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

Original proposal:
Have a complete OpenAPI compliant documentation for the Open Food Facts API ...

Expected outcomes:

Refresh the Open Food Facts API and make it OpenAPI compliant

- Audit the existing documentation and create a friction log, especially around the API itself, and the tools used to maintain it.
- Refresh the Open Food Facts API based on feedback/exchanges with the community
- Extend the documentation with all the new routes and APIs listed in https://github.com/openfoodfacts/api-documentation
- Ensure all existing routes are compliant with best practices for OpenAPI that allow for SDK generation, testing...
- Participate in leveling up our API game and best practices, along with the core Open Food Facts team

- (stretch goal) Set up a GitHub Actions CI pipeline with mentors to convert the doc to Open API (and then to OpenAPI generated SDKs)

Estimated time on the project: 200 hours (25 days on 5 month).

We have identified mentors from the core as well as volunteers from the community.

We have not yet identified candidates for the projects. We hope to get some applications from GSOC but we also have the ability to broadcast the project proposal to our wider community of users.

Measuring your project's success

Open Food Facts is reused by more than 160 mobile apps thanks to our current API. The more it is clear and efficient, the more we gain new reusers, and the less support we have to provide (which is time consuming). We're also facing the need to gather more apps to reuse the write API to send back products or data to the Open Food Facts database.

We would consider the project successful if, after publication of the new documentation:

- The number of API users increases by 10% in the following year.
- Within a year, at least 5 additional users of the API are now sending back products to the Open Food Facts database.
- At least 3 SDK are generated thanks to OpenAPI (we will focus on widely adopted languages: Python, Javascript, …)
- The API questions are mostly bug or corner cases related and not linked to miscomprehensions of the API

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?

The decision was quickly made, since the API is the most important piece of documentation we have, and that it provokes the bulk of our systemic impact by empowering others to build with Open Food Facts. Several members of the core team wrote the proposal on Google Docs and Google Spreadsheets, and we reviewed it over a few days.

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?

Budget for the "Have a complete OpenAPI compliant documentation for the Open Food Facts API"

- Technical writing: 25 days @$350/day — $8750
- Management/technical coaching: 2 days @$400/day — $800
- Dev ops (pipeline, deployment, tests…): 3 days @$400/day — $1200

Total: 10750

We stuck to the budget.

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?

We published Have a complete OpenAPI compliant documentation for the Open… on the Google Season of Docs website and on our social media, and selected candidates based on their application. We had an interview for the most promising candidates.

- Funke Olasupo - technical writer, was the sole paid contributor.
- Stéphane Gigandet - lead developer - Open Food Facts core team
- Alex Garel - developer - Open Food Facts core team

- Pierre Slamich - product manager - Open Food Facts core team

Note: We did not realize we could spread the project over several contributors

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?

Original timeline

The project itself will take approximately five months to complete. Once the tech writer is hired, we'll spend two weeks on tech writer orientation, then move onto the audit and friction log, and spend the last few months focusing on fixing the documentation and moving it into the OpenAPI standard. We will closely follow the proposed GSOC timeline

https://developers.google.com/season-of-docs/docs/timeline

Dates	Action Items
May	Orientation
June	Audit existing documentation and create friction log
July - August	Fixing current documentation and work on open issues
September - October	Moving to OpenAPI and Devops

Revised timeline

We had to keep the timeline reachable by removing some not so essential parts of the documentation. The reasons are outlined below in the analysis section.

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?

In first part of the project, Funke created a friction logs, after interviewing several external developers and the core team: https://github.com/openfoodfacts/api-documentation/tree/develop/friction-log

The main result is a completely new OpenAPI documentation. This documentation is still rough on the edges and needs to be refined on some part. However it provides the documentation for the most important points and is already more precise than the old documentation.

The published documentation: https://openfoodfacts.github.io/openfoodfacts-server/reference/api/

The source file: https://github.com/openfoodfacts/openfoodfacts-server/tree/main/docs/reference/api.yml

This documentation has the advantage that it leaves in the main repository. We now can insure that a PR which change the API can add the corresponding documentation change.

Another important work is the writing of an introduction and a tutorial. The API is quite hairy and not always very easy, so the reference documentation is not enough. The tutorial will hopefully smooth the learning curve for newcomers.

We also reworked the documentation menu to help discoverability.

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 number of API users increases by 10% in the following year.
    - we monitor API usage yearly, although we are currently adding live metrics to be able to monitor it more closely
- Within a year, at least 5 additional users of the API are now sending back products to the Open Food Facts database.
    - in parallel to the project, we started an official open inventory of applications re-use

    - also our 10 years anniversary event was the occasion to meet some of our re-users
    - we monitor re-use in our yearly report
- At least 3 SDK are generated thanks to OpenAPI (we will focus on widely adopted languages: Python, Javascript, …)
    - This is monitored manually, as normally such repos are publicized on our slack. Right now we don't have any project.
- The API questions are mostly bug or corner cases related and not linked to miscomprehensions of the API
    - We don't have metrics on this, but the burden is
    - At the moment, this kind of questions are still there but it's easier to answer half of them by sending a quick link to our tutorial.

It's still too early to measure those metrics, but the doc is published, with a strong process and feedback loop, points 3 and 4 are already promising, and 1 and 2 will take more time to be assessed.

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.)

We consider that - although reduced in documentation scope - the project is successful.

We believe that the project is successful in that it has created a culture of using OpenAPI across the various Open Food Facts projects, an automated publication loop, and internal weekly processes and the bundling of documentation within project, enabling PRs to contain code and API documentation.

This move to OpenAPI also encourage having or enhancing the OpenAPI of other projects like robotoff, taxonomy-editor, facets knowledge panels, and folksonomy engine.

Unexpected things:

- We realized the difficulty of using OpenAPI on an API that was not made for that, which led us to use OpenAPI v3.1 which is really more expressive thanks to full JSON schema compatibility, but which also cuts us off from quite a few Open Source tools that have very limited support for v3.1
- The research for tools took quite a long time. OpenAPI specification is quite intimidating for a technical writer and many online resources are for older

versions of OpenAPI. Finding good tools to help make first steps was really important.

- API documentation helps us rethink how we wanted the API for future versions. The contact with OpenAPI specification, also helps this process by exposure to good API practices. The v3 API of openfoodfacts server has already been started using OpenAPI, leveraging the current documentation.
- We realized how much OpenAPI requires real dev skills which are sometimes a bit complex for a technical writer.
- Devs had to create part of the documentation as a result
- It's not easy to write a doc when the only existing up to date doc is the PERL code :-) Therefore, details are still missing. It motivates the dev team to try to take more time regularly to cope with that.
- The effectiveness and expressiveness of OpenAPI gives devs the taste to write more documentation.

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?

We learned a great deal about OpenAPI and modern API documentation writing, while mentoring Funke.

We would potentially hire several technical writers as suggested in the guidelines, and potentially with a programming/OpenAPI background, possibly in Perl (or whatever language involved)

Projects should make a conscious decision to hire either a seasoned OpenAPI programmer able to work with reduced input, or to mentor a Technical Writer into the codebase and toolchain.

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.

- Stoplight Studio is a tool that has proved very useful to documentation writing, also thanks to its good support of OpenAPI 3.1
- OpenCollective helps us have payments reach their recipient in a difficult context