

Winter of Code 5.0

Proposal

for the Project

“Scan API”

under



CUMBUCA DEV

About Me

Name - Suryam Singh

Email-Id - suryam1802singh@gmail.com

GitHub Username - kaemi-kun

Country - India

Timezone - IST (Indian Standard Time) — UTC +05:30

Primary Language - English

LinkedIn-Profile-Link -

https://www.linkedin.com/in/suryam-singh-626061346?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=android_app

Other Links (If Applicable) - github.com/kaemi-kun

Synopsis

APIs are a core part of modern applications, but testing them properly and keeping their behavior aligned with documentation is still a challenge for many teams. ScanAPI solves this by allowing developers to test APIs directly from OpenAPI specifications. However, in real projects, API specifications often include edge cases, unclear errors, and complex schema definitions that are not always easy to validate or understand.

This project focuses on making ScanAPI more friendly and reliable for everyday developers. The work will improve support for real-world OpenAPI scenarios, make error messages easier to understand, add more automated tests, and improve documentation and examples. As a highlight contribution, the project will also improve how ScanAPI handles **oneOf** and **anyOf** schema definitions, which are common in modern APIs but often confusing to validate correctly.

Overall, the goal is to make ScanAPI easier to use, easier to debug, and more trustworthy for developers who rely on automated API testing in their daily workflows.

Benefits to the Community

This project helps ScanAPI by making the tool more reliable and easier to use in real-world API testing workflows. By improving support for common OpenAPI edge cases, expanding test coverage, and making error messages clearer, ScanAPI becomes more practical for developers who rely on it for validating their APIs. Better documentation and examples will also reduce confusion for new users and make onboarding smoother.

The highlight improvement in handling `oneOf` and `anyOf` schemas allows ScanAPI to support more realistic and complex API specifications, increasing its usefulness for modern applications. Overall, this contribution strengthens ScanAPI's reputation as a dependable, developer-friendly API testing tool and supports its long-term growth within the open-source and professional developer communities.

Project Plan

I plan to approach this project in small, practical steps so that each improvement is easy to review, test, and refine. I will begin by setting up ScanAPI locally and running it against a few real OpenAPI specifications to understand how the tool behaves in common and edge-case scenarios. This will help me identify areas where validation fails, error messages are unclear, or documentation does not fully explain the behavior.

After this exploration phase, I will focus on improving existing features rather than introducing entirely new systems. My first contributions will target documentation and examples, making sure that new users can easily understand how to write specifications, interpret results, and debug failures. In parallel, I will add automated tests for common OpenAPI edge cases so that ScanAPI can handle more real-world specifications reliably.

Next, I will work on improving error reporting by making failure messages more descriptive and helpful. Instead of generic errors, I will aim to provide clearer context about what failed, where it failed, and why it failed. This will make debugging much easier for developers using ScanAPI in CI or daily workflows.

As the highlight part of the project, I will improve ScanAPI's handling of `oneOf` and `anyOf` schema definitions. I will first study how these cases are currently processed, then extend support for a practical subset of scenarios. Each change will be supported by automated tests and documented clearly so that users understand the expected behavior.

Throughout the project, I will regularly seek feedback from mentors, refine my contributions based on reviews, and ensure that all changes follow ScanAPI's contribution and coding guidelines.

Tech Stack

- Python
- Pytest
- OpenAPI (YAML / JSON)
- HTTP client libraries used by ScanAPI
- GitHub Actions for CI

Milestones

Milestone	Tentative Date	KPI
Week 1	1/02/26 - 7/02/26	Set up ScanAPI locally, run it on multiple OpenAPI specs, and note common failures, unclear errors, and documentation gaps
Week 2	8/02/26 - 14/02/26	Improve documentation and add clearer examples for writing specs and understanding test results
Week 3	15/02/26 - 21/02/26	Add automated tests for common OpenAPI edge cases such as missing fields, optional parameters, and unexpected responses
Week 4	22/02/26 - 28/02/26	Improve error messages to include clearer context about what failed and why
Week 5	1/03/26 - 7/03/26	Improve support for <code>oneOf</code> / <code>anyOf</code> schema handling for a practical subset of cases, with tests
Week 6	8/03/26 - 14/03/26	Refine contributions based on mentor feedback, improve documentation, and

		finalize testing.
--	--	-------------------

Deliverables

By the end of the contribution period, I aim to deliver:

- Improved documentation and examples that make it easier for new users to write specifications and understand ScanAPI results.
- Additional automated test cases covering common OpenAPI edge scenarios such as missing fields, optional parameters, and unexpected responses.
- Clearer and more helpful error messages that explain what failed and why in a user-friendly way.
- Enhanced support for **oneOf** and **anyOf** schema handling for a practical subset of real-world cases, along with corresponding tests
- Updated test coverage to ensure new and existing features remain stable.
- Minor code cleanups and readability improvements in the areas touched during the project.

Acknowledgement

I sincerely thank the Scan API maintainers and the Winter of Code team for creating this opportunity to learn, contribute, and grow through open source development. I look forward to collaborating with the mentors and community and contributing meaningfully to the project.

Thank you!!