Documentation

Index

Chapter 1 - Things you should know beforehand

A. What is OPEN API specification?

OpenAPI Specification (formerly Swagger Specification) is an API description format for REST APIs. An OpenAPI file allows you to describe your entire API, including:

- Available endpoints (/users) and operations on each endpoint (GET /users, POST /users)
- Operation parameters Input and output for each operation
- Authentication methods
- Contact information, license, terms of use and other information.

API specifications can be written in YAML or JSON. The format is easy to learn and readable to both humans and machines. The complete OpenAPI Specification can be found on GitHub using the following URL : https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md

B. What is Swagger?

Swagger is a set of open-source tools built around the OpenAPI Specification that can help you design, build, document and consume REST APIs. The major Swagger tools include:

- Swagger Editor – browser-based editor where you can write OpenAPI specs.
- Swagger UI – renders OpenAPI specs as interactive API documentation.
- Swagger Codegen – generates server stubs and client libraries from an Open API spec.

C. Why use OPEN API ?

The ability of APIs to describe their own structure is the root of all awesomeness in OpenAPI. Once written, an OpenAPI specification and Swagger tools can drive your API development further in various ways:

- Design-first users: We use Swagger Codegen to generate a server stub for your API. The only thing left is to implement the server logic and your API is ready to go live!
- Use Swagger Codegen to generate client libraries for your API in over 40 languages.
- Use Swagger UI to generate interactive API documentation that lets your users try out the API calls directly in the browser.
- Use the spec to connect API-related tools to your API. For example, import the spec to SoapUI to create automated tests for your API.
- Many open-source and commercial tools can be integrated with Swagger.

Chapter 2 - Features of Swagger Hub

A. Definition Editor

The Definition Editor comprises of three important sections namely Edit, Validate and Collaborate.

Edit :- Here is where the API definition is built so it is one of the most critical components of Swagger Hub. The editing is done on swagger-editor which is an open source tool. With Editor one can build API definition in YAML editor and can be seen coming to life in the visualizer panel

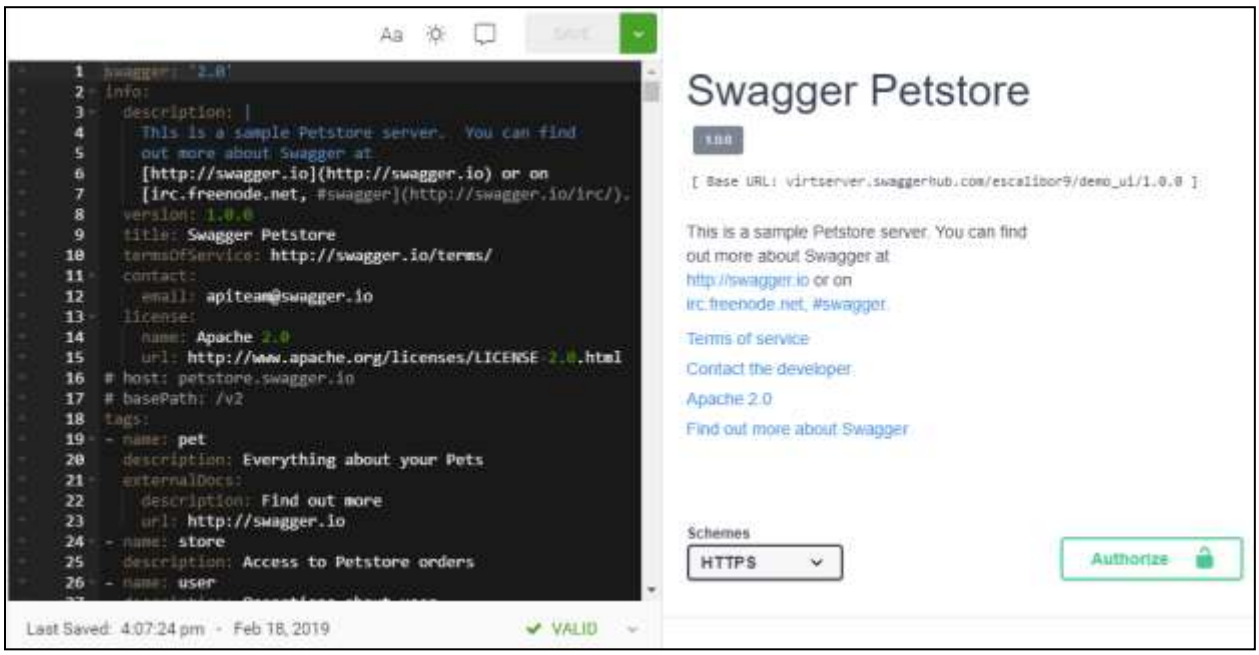

Figure 1: The Left Pane denotes the Swagger Editor where defining of the document is done. The right pane is the visualizer where the API definition comes to life. The "VALID" checkbox is green if there is no syntactical or indentation error.

Validate :- One of the most common requests from people is to provide them with timely and useful error messages. Whatever definition is build is compiled in real time and it is ensured that

the definition meets the Swagger specification.

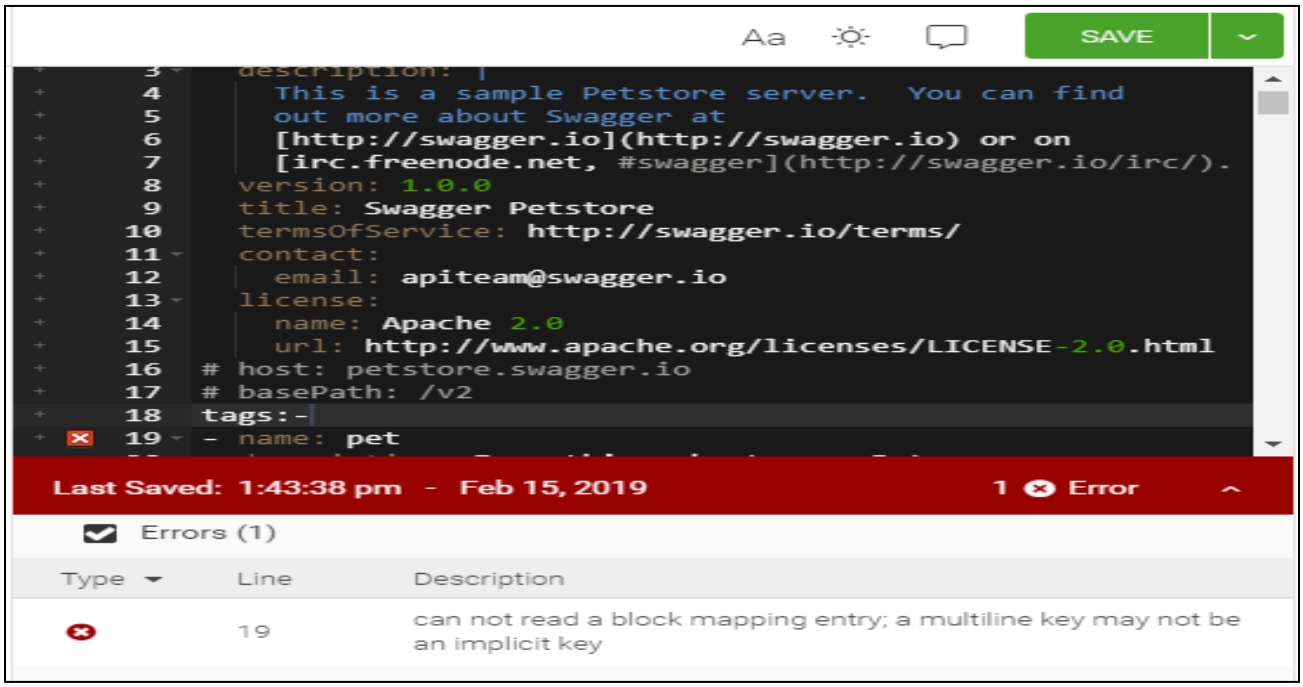

Figure 2: Shows the line number as well as type of error occurred

Collaborate :- This feature is used to invite other API developers to work on an API definition with you. One can invite up to 25 people. Collaborators can revise API definitions without restriction, allowing development teams to work efficiently and without a lot of administrative overhead.

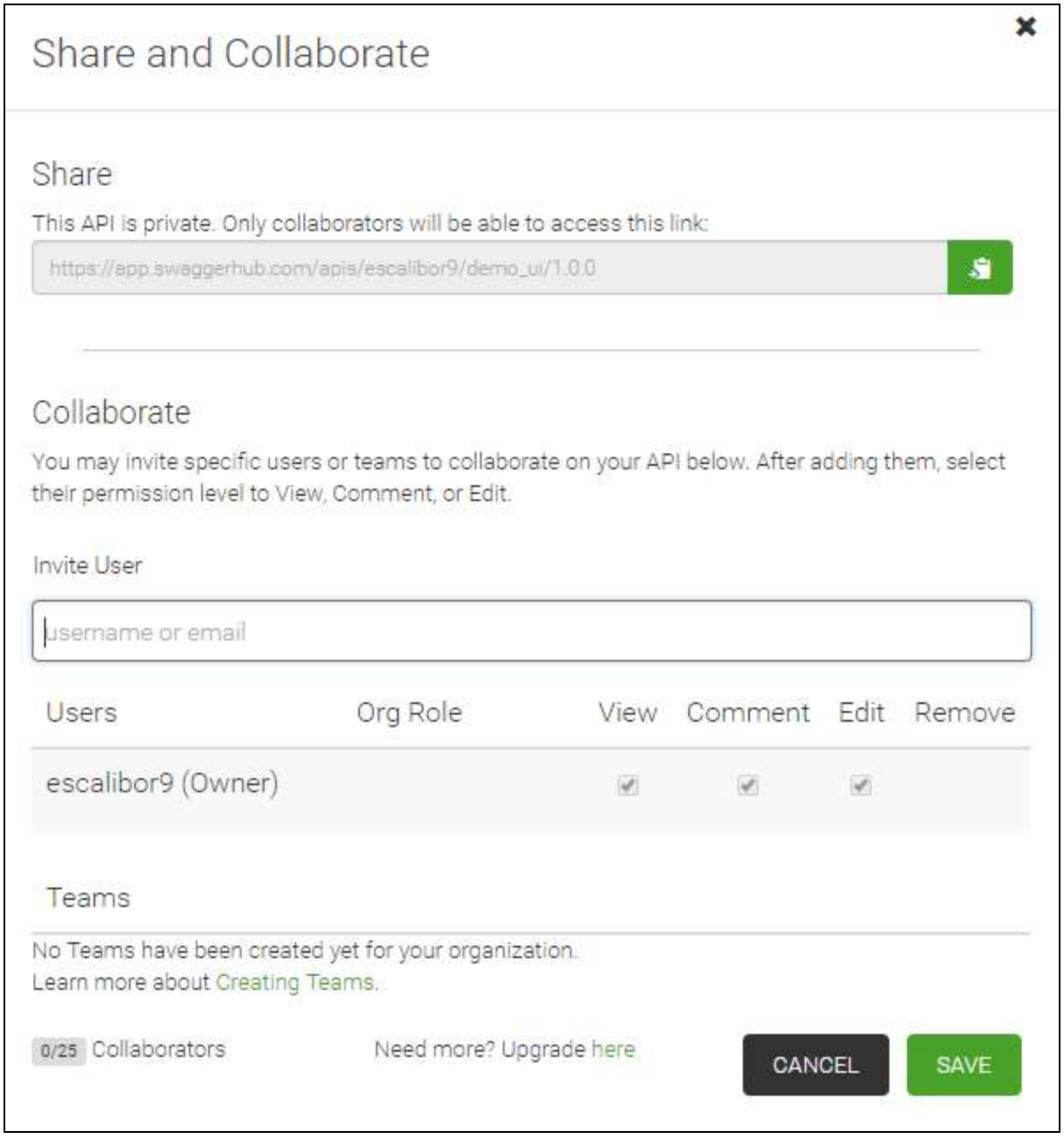

Figure 3: Here Collaborators can be added and action given to each collaborator.

B. Code Gen

After collaboration, the API definition can now be taken and turned into client SDKs or server-side code quickly and easily using SwaggerHub's code templates.

Client: - With 26 client code templates, SwaggerHub makes it as simple as choosing from a drop-down list.

Server: - SwaggerHub's code generation capabilities are not limited to client libraries. They provide a full set of the most popular server-side code templates as well, all of which use the mustache template structure, making it very easy to write static documentation, supporting classes, test harnesses, etc. They also include the associated POM file.

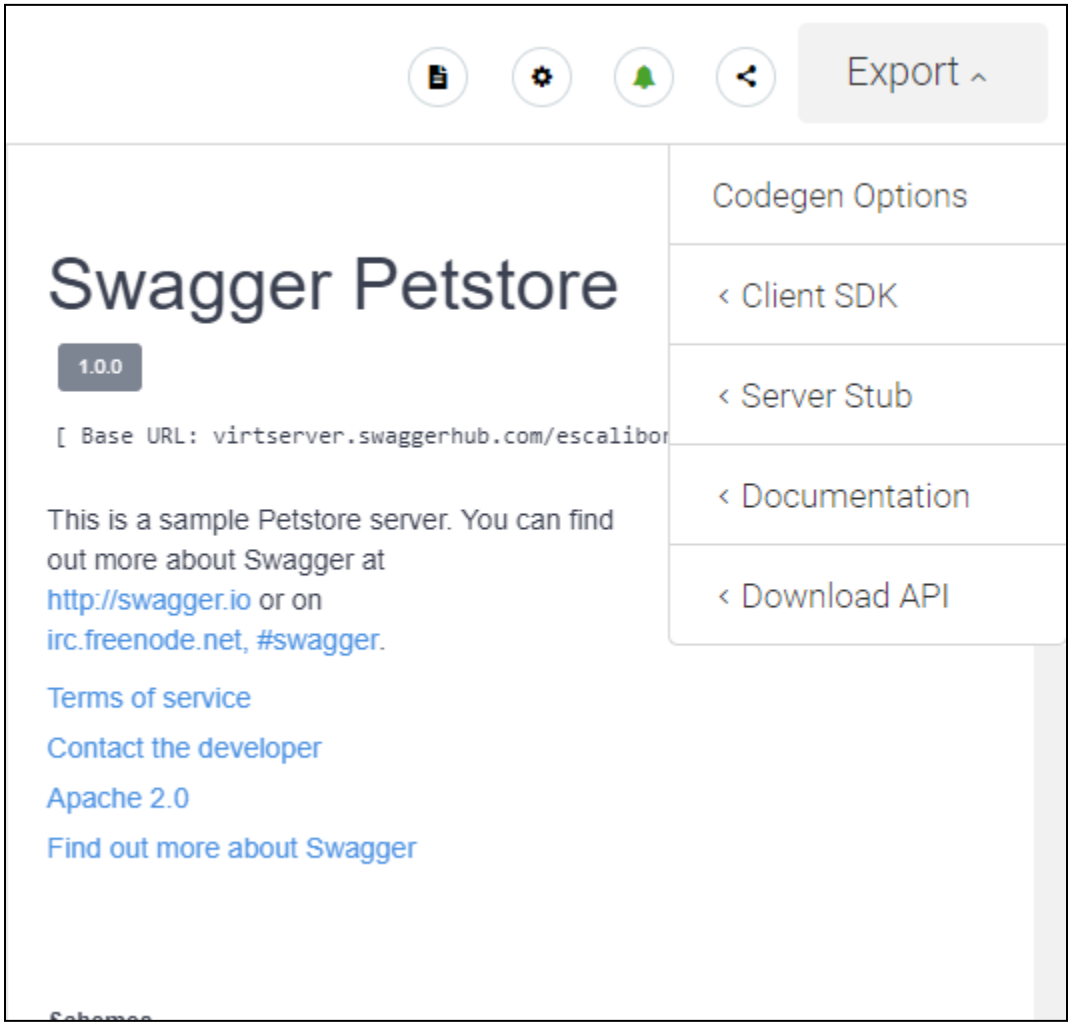

Figure 4: Generates the API definition into both client and server side from a list of options available.

C. Versioning

Versioning: - One can have up to 25 versions of API definition in SwaggerHub. This allows continually refinement and enhancement of definitions while still having a clean published version for API consumers to see.

For example, if you are working with collaborators on the next version of your API definition, you might want to create a collaboration version for everyone to work on together while maintaining a solid state version where you can merge the changes you want to incorporate.

There are multiple ways to version API definition:

- Save an unpublished definition as another version.
- Bump a published definition to a new version after you make changes to it (this preserves the published version).
- Fork to a new API/version altogether.

Publish/ Unpublish: - When the created API definition is ready for others to see in the SwaggerHub Registry, one can publish it. Publishing an API definition effectively locks it so any changes you make after that point will be saved as a different version. If collaborators are added during the course of building that definition, they won't be able to make any changes to the published version either.

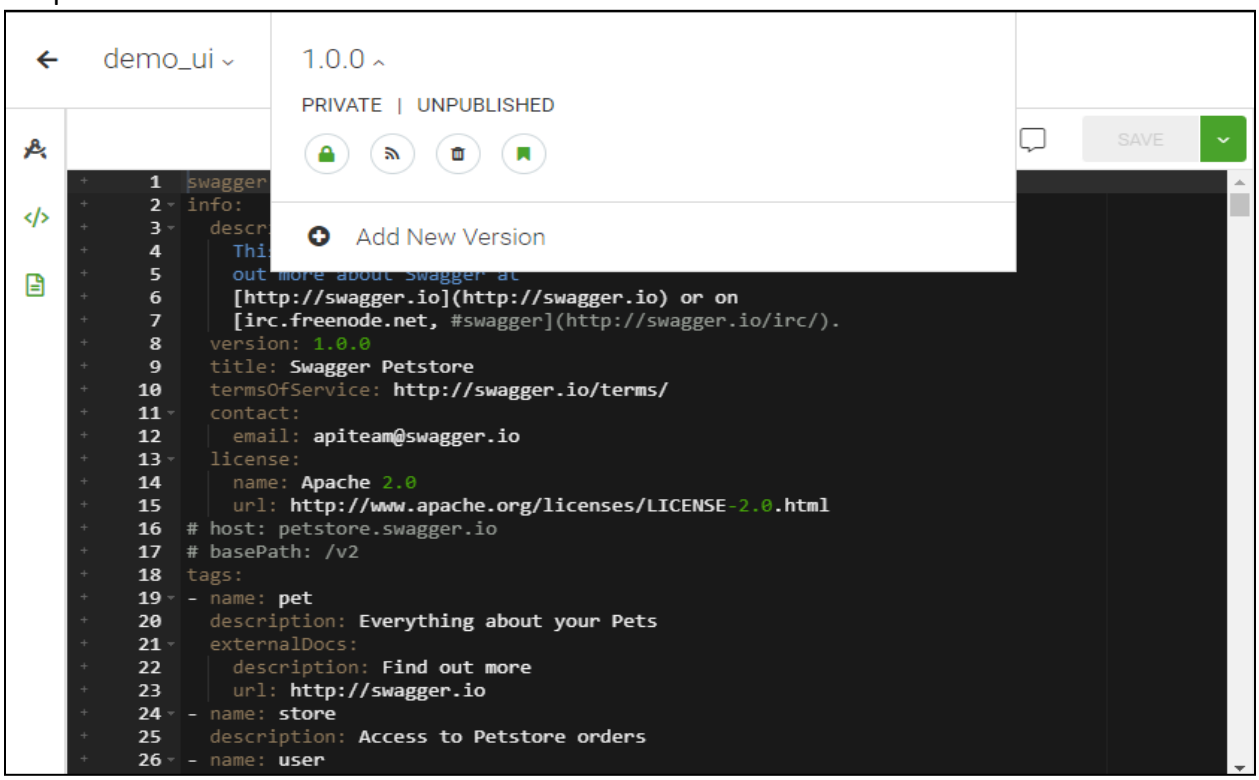

Figure 5: New versions can be added and APIs can be published/unpublished.

If one needs to pull that API definition back to make some fixes, it can be Unpublished. Unpublishing an API definition removes it from the SwaggerHub registry so one can continue to modify it without all that public scrutiny.

Push to GitHub: - The API definitions can be pushed to your repository with the rest of your assets. So, after you've merged the contributions from your collaborators and you feel you have something ready to commit, use the Push to GitHub option to commit the current version of the API definition to your private or shared repository.

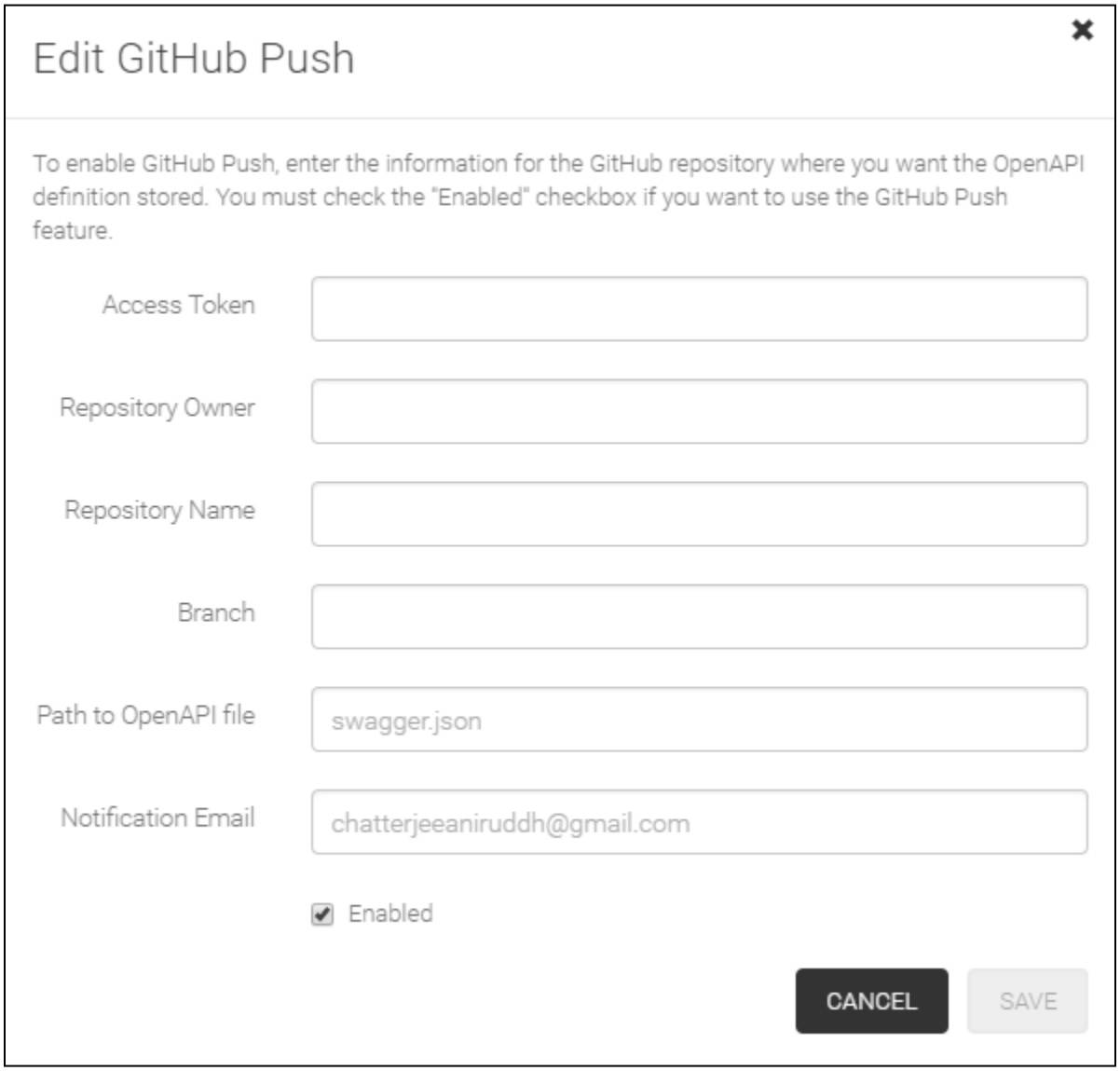

Figure 6: Versioning can also be maintained by pushing into your GitHub repository.

D. API Registry

One can find and explore APIs without much effort. SwaggerHub puts the API definition at the center of the development process making API Registry available for everyone.

Search: - SwaggerHub provides a comprehensive listing of Swagger-based APIs. To see the full list, just leave the Search box empty and click the Search icon. You can sort the list by name or the date the API definition was last updated.

Explore: - When one finds the API one is interested in on the search list, SwaggerHub lets API providers manage multiple versions of their API on the registry. Flip between versions by using the version drop-down next to the API name and open up any of those versions to explore further.

Speaking of exploring further, an interactive document interface allows to see exactly what the API can and even try it out.

Use: - After downloading the API's YAML or JSON by simply selecting which you want from the "Download" drop-down, one can link it to DockerHub. Clicking the icon will bring you directly to DockerHub so you can download the container associated with the API.

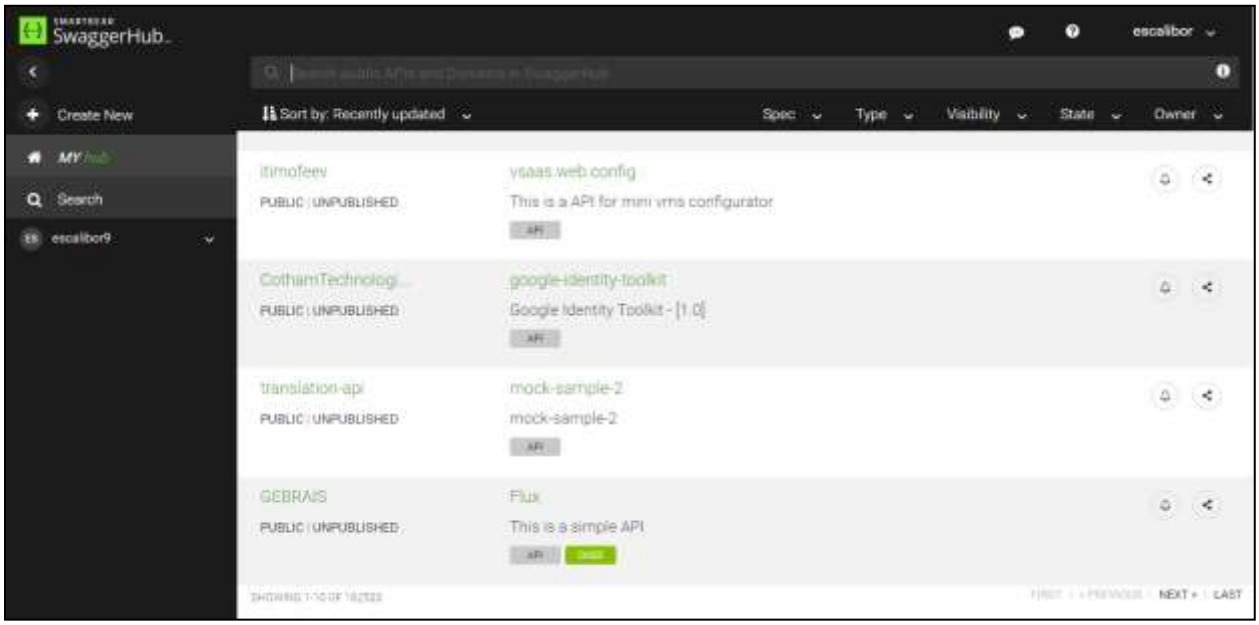

Figure 7: All the published APIs can be searched, explored and used using the Search bar.

E. Domains

Designing and developing APIs involve a good amount of repeating and re-writing descriptions.

Domains are a collection of reusable components. The components that can be stored inside a Domain are:

- **Definitions** These are models which describe your APIs inputs and outputs
- **Path Items** A set of predefined operations that can be reused across paths
- **Parameters** Any input into an API
- **Responses** The output from making an API call

Users can create and version Domains, and then define the reusable components that can be stored inside them. The components can be referenced from other APIs or Domains, either by the user or the collaborators on the API. The Domain UI helps all the API stakeholders better visualize the components of the Domain, while allowing for quick readability and detection of errors.

Creating a domain: -

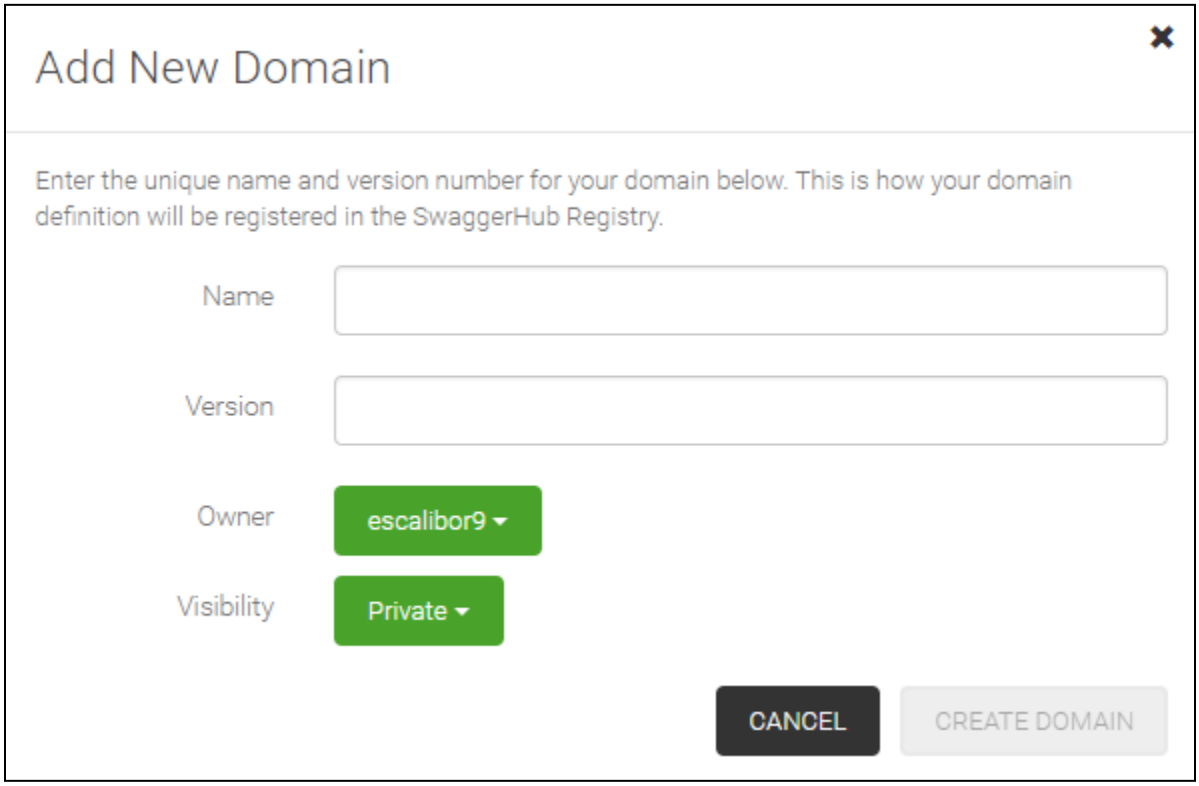

Figure 8: Creating a new domain is similar to creating a new API.

Managing a domain: - Domains are managed the same way as APIs. Created Domains can be edited, bumped to a different version, pushed to GitHub or deleted. Domains can be either Public or Private. Public Domains can be searched for, viewed, referenced and forked by other SwaggerHub users, while Private Domains can only be viewed and worked on by you and the collaborators you've added.

Domains can also act as a control center for multiple APIs; one change in a Domain can quickly transmit across all the APIs which reference the Domain, thus allowing for faster API development and collaboration.

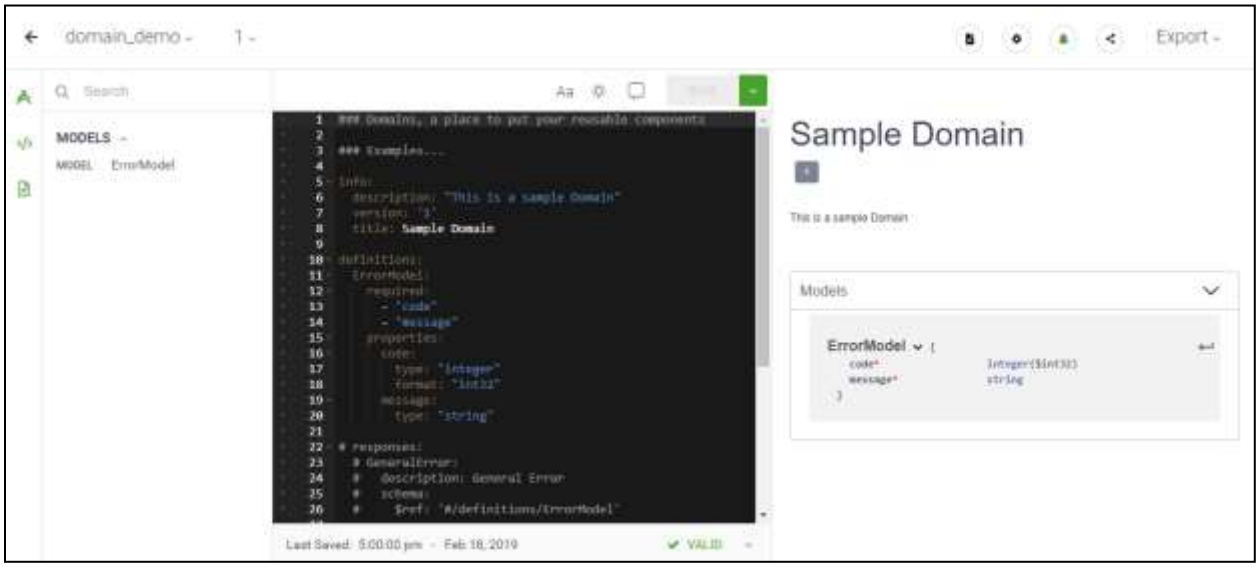

Figure 9: An editor and a visualizer pane after creating a sample domain.

F. Plugins

Plugins are a neat way to drive your API lifecycle with connections into your existing toolset. These integrations are bits of software that can be added to your API definition to expand SwaggerHub's functionality under your control.

Some mostly used integration is – GitHub sync and Webhooks.

GitHub Sync: - GitHub Sync allows generated code to be automatically synchronized with your GitHub repository. That means, no more downloads, merging code, complicated configurations, or scripting. Simply specify the target to generate, the GitHub repository information, and file/folder information and your clients and servers will be always be up-to-date with your Swagger definition. Clients, servers alike can be pushed to one or many repositories.

Webhooks: - Webhooks allow you to trigger your own services and integrations during the lifecycle events that you are interested in. When triggered, the Swagger definition is pushed to the service you define for a number of events. Whether triggering a build, a unit test, emailing

your developers, or other, you can dream up automation scenarios with SwaggerHub Webhooks.

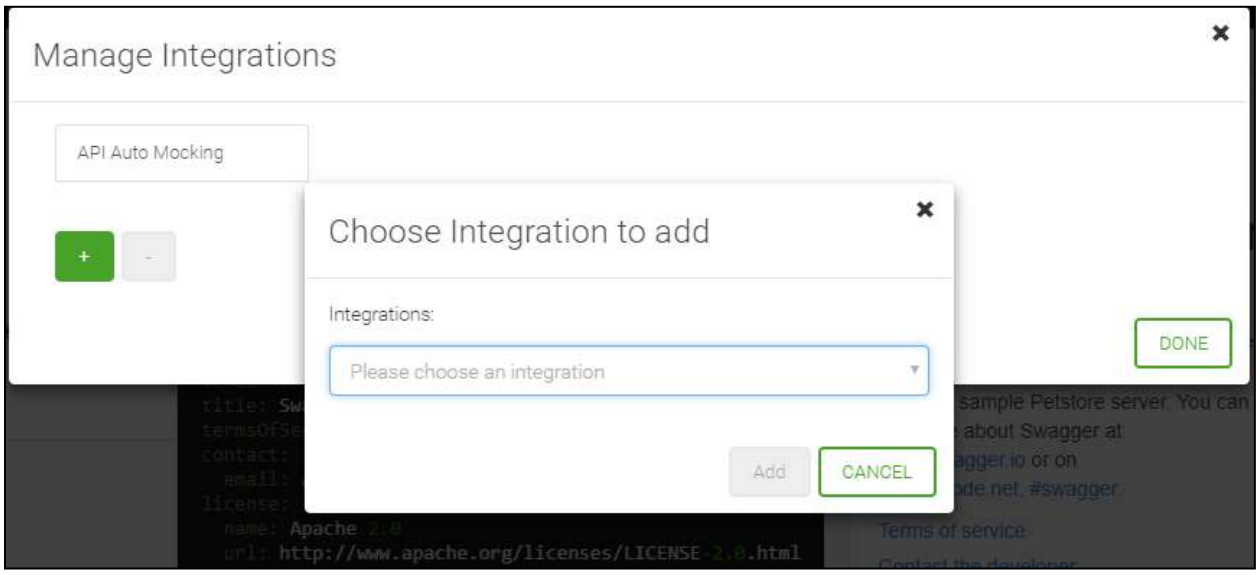

Figure 10: From the drop down list, the desired integrations can be chosen.

Chapter 3 – Testing & Documentation

A. What is Swagger Inspector?

Swagger Inspector is an online web application for developers, testers, product managers and whoever interested to test APIs and to generate documentation about the APIs.

Swagger Inspector Features:

- It supports all the HTTP methods.
- You can pre-fill the data to try the features. It supports file upload as well.
- It stores all the requests in the History.
- You can create Collections for your APIs.
- It has simple and elegant UI.

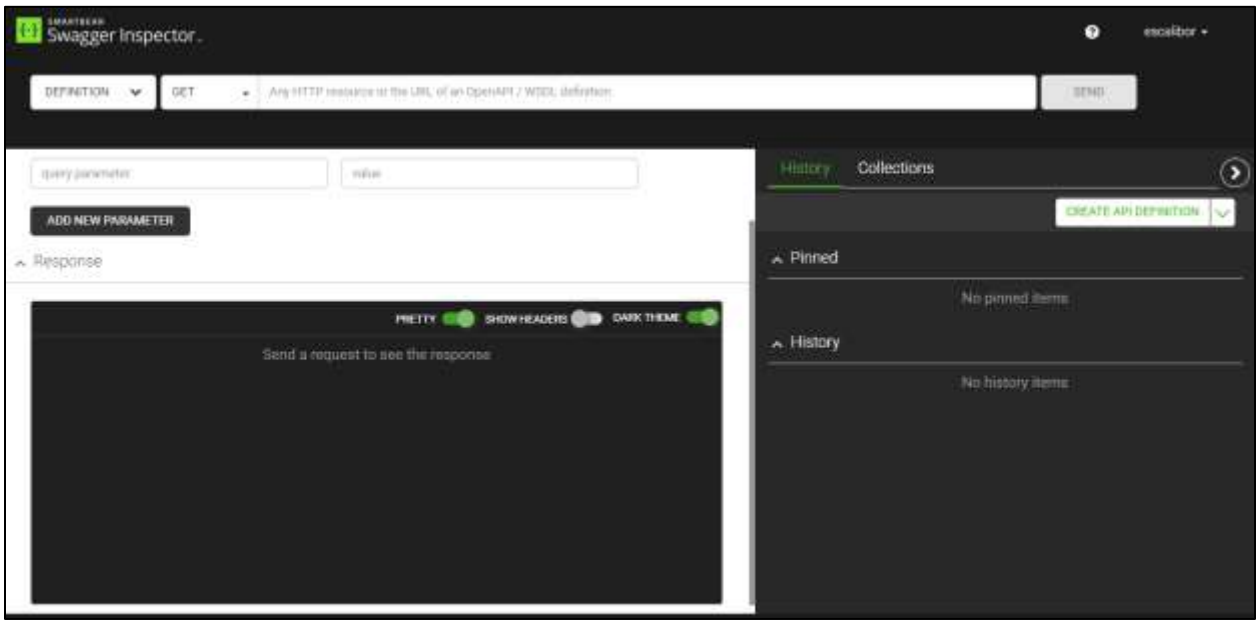

Figure 11: The Swagger Inspector UI.

Swagger Inspector Functionalities:

- Test created APIs
- Mock API Calls
- Document APIs