

Running a Web API and using Git (with OpenAPI – Swagger)

1 Introduction

1.1 OpenAPI

The official name for the OpenAPI specification. The OpenAPI specification provides a set of properties that can be used to describe your REST API. When valid, the specification document can be used to create interactive documentation, generate client SDKs, run unit tests, and more. You can read the specification details on GitHub at <https://github.com/OAI/OpenAPI-Specification>. Under the Open API Initiative with the Linux Foundation, the OpenAPI specification aims to be vendor neutral.

1.2 Swagger

Refers to API tooling related to the OpenAPI spec. Some of these tools include Swagger Editor, Swagger UI, Swagger Codegen, SwaggerHub, and others. These tools are managed by Smartbear. “Swagger” was the original name of the OpenAPI spec, but the name was later changed to OpenAPI to reinforce the open, non-proprietary nature of the standard. People sometimes refer to both names interchangeably (especially on older web pages), but “OpenAPI” is how the spec should be referred to.

2 Swagger tutorial

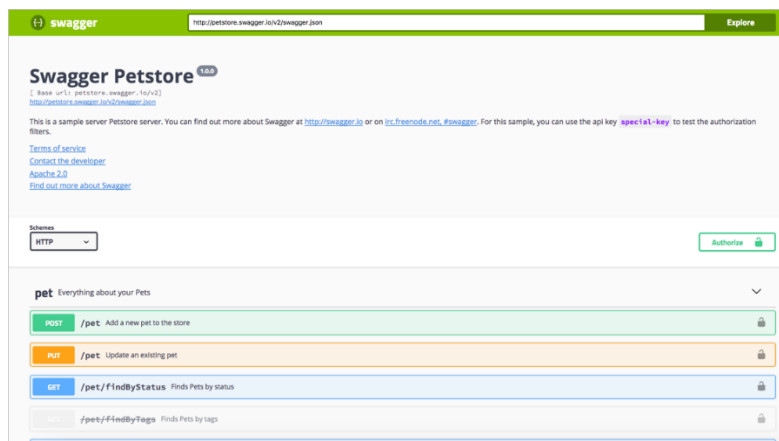
Swagger UI (<https://github.com/swagger-api/swagger-ui>) provides a display framework that reads an OpenAPI specification document (<https://github.com/OAI/OpenAPI-Specification>) and generates an interactive documentation website. The following tutorial shows you how to integrate an OpenAPI specification document into Swagger UI.

2.1 Swagger UI overview

Swagger UI is one of the most popular tools for generating interactive documentation from your OpenAPI document. Swagger UI generates an interactive API console for users to quickly learn about your API and experiment with requests. Additionally, Swagger UI (which is an actively managed project with an Apache 2.0 license) supports the latest version of the OpenAPI spec (3.x) and integrates with other Swagger tools.

2.2 Get familiar with Swagger UI through the Petstore demo

To get a better understanding of Swagger UI, let's explore the Swagger Petstore example (<http://petstore.swagger.io/>). In the Petstore example, the site is generated using Swagger UI.



The endpoints are grouped as follows:

- [pet](#)
- [store](#)
- [user](#)

2.2.1 Authorize your requests

Before making any requests, you would normally authorize your session by clicking the **Authorize** button and completing the information required in the Authorization modal.

The Petstore example has an OAuth 2.0 security model. However, the authorization code is just for demonstration purposes. There isn't any real logic authorizing those requests, so you can simply close the Authorization modal.

2.2.2 Make a request

Now let's make a request:

1. Expand the [POST Pet](#) endpoint.
2. Click **Try it out**.
3. After you click Try it out, the example value in the Request Body field becomes editable.
4. In the Example Value field, change the first id value to a random integer, such as 193844. Change the second name value to something you'd recognize (your pet's name).
5. Click **Execute**.

Swagger UI submits the request and shows the `curl` that was submitted. The Responses section shows the response. (If you select JSON rather than XML in the "Response content type" drop-down box, the response's format will be shown in JSON.)

The Petstore is a functioning API, and you have actually created a pet.

2.2.3 Verify that your pet was created

1. Expand the [GET /pet/{petId}](#) endpoint.
2. Click **Try it out**.
3. Enter the pet ID you used in the previous operation. (If you forgot it, look back in the **POST Pet** endpoint to check the value.)
4. Click **Execute**. You should see your pet's name returned in the Response section.

2.2.4 Some sample Swagger UI doc sites

Before we get into this Swagger tutorial with another API (other than the Petstore demo), check out a few Swagger implementations:

- [Reverb \(https://reverb.com/swagger#/articles\)](https://reverb.com/swagger#/articles)
- [VocaDB \(http://vocadb.net/swagger/ui/index\)](http://vocadb.net/swagger/ui/index)
- [Watson Developer Cloud \(https://watson-api-explorer.mybluemix.net/\)](https://watson-api-explorer.mybluemix.net/)
- [The Movie Database API \(https://developers.themoviedb.org/3/account\)](https://developers.themoviedb.org/3/account)
- [Zomato API \(https://developers.zomato.com/documentation\)](https://developers.zomato.com/documentation)

Some of these sites look the same, but others, such as The Movie Database API and Zomato, have been integrated seamlessly into the rest of their documentation website.

2.3 Create a Swagger UI display with an OpenAPI spec document

In this activity, you'll create a Swagger UI display for an OpenAPI specification document. If you're using one of the pre-built OpenAPI files, you can see a demo of what we'll build here:

OpenWeatherMap Swagger UI

(https://idratherbewriting.com/learnapidoc/pubapis/swagger_demo.html).

First, you'll make sure you can view Swagger locally. Then you'll switch the Petstore OpenAPI document URL with an OpenWeatherMap OpenAPI document URL.

1. Go to the Swagger UI GitHub project. (<https://github.com/swagger-api/swagger-ui>)
2. Click **Clone or download**, and then click **Download ZIP**. Download the files to a convenient location on your computer and extract the files.

The only folder you'll be working with in the downloaded zip is the `dist` folder (short for distribution). Everything else is used only if you're recompiling the Swagger files, which is beyond the scope of this tutorial. (If desired, you can drag the `dist` folder out of the `swagger-ui-master` folder so that it stands alone.)

3. In your Chrome browser, press **Cmd+O** (Mac) or **Ctrl+O** (Windows), browse to the `dist` folder, and select the `index.html` file,

You should see the Petstore Swagger content. Now you'll customize the OpenAPI spec file with another file.

4. Inside your `dist` folder, open `index.html` in a text editor such as Visual Studio Code.
5. Look for the following code:

```
url: "http://petstore.swagger.io/v2/swagger.json",
```

6. Change the url value to an online web URL to your Swagger file. For example: an [OpenWeatherMap OpenAPI specification](https://idratherbewriting.com/learnapidoc/docs/openapi_spec_and_generated_ref_docs/openapi_openweathermap.yml) (https://idratherbewriting.com/learnapidoc/docs/openapi_spec_and_generated_ref_docs/openapi_openweathermap.yml) Then save the file.

If the url reference isn't to an online URL, Swagger UI will create an CORS (cross-origin resource sharing) error. To view Swagger UI with a local OpenAPI file, you can run a simple Python server locally

(<https://docs.python.org/2/library/simplehttpserver.html>) to simulate a web server (this requires you to install Python).

7. Refresh the `index.html` file in your Chrome browser. The content should show the OpenWeatherMap API content instead of Swagger Petstore content.

When you're ready to publish your Swagger UI file, you just upload the `dist` folder (or whatever you want to call it) to a web server and go to the `index.html` file. For example, if you called your directory `dist` (leaving it unchanged), you would go to `http://myserver.com/dist/`.

For more instructions in working with Swagger UI, see the Swagger.io docs

(<https://swagger.io/docs/open-source-tools/swagger-ui/usage/installation/>).

2.4 Activity: View local OpenAPI file in Swagger UI

In order to view a *local* OpenAPI file (rather than an OpenAPI file hosted on a web server), you'll need to run an HTTP server on your computer. This is because CORS (cross-origin resource

sharing) security restrictions in Chrome will block Swagger UI from running. Swagger UI needs to load on a web server to fulfill the security requirements.

You can create a local web server running on your computer through Python's [SimpleHTTPServer module](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/set_up_a_local_testing_server) (https://developer.mozilla.org/en-US/docs/Learn/Common_questions/set_up_a_local_testing_server).

Mac has a system version of Python installed by default, but Windows computers will need to install Python.

Windows: Run the Python simple HTTP server

1. Download and install Python 3.9.1 (<https://www.python.org/downloads/>)

When you install Python, be sure to select the check box that says "Add Python 3.9.1 to PATH." *This check box isn't selected by default. If you don't select it, your command prompt won't recognize the word "python".*

Add Python 3.9.1 to PATH

2. After installing Python, close your command prompt and reopen it.
3. In your command prompt, browse to the Swagger UI `dist` directory.

To browse in the Windows command prompt, type `cd <folder name>` to move into the folder. Type `cd ..` to move up a directory. Type `dir` to see a list of the current directory's contents.

If you're having trouble locating the `dist` directory in the command prompt, try this trick: type `cd`, press the spacebar, and then drag the `dist` folder directly into the command prompt. The path will be printed automatically.

4. After you've navigated into the `dist` folder, launch the server:

```
python3 -m http.server
```

If this command doesn't work, try it without the 3:

```
python -m http.server
```

The server starts:

```
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

(If your Command Prompt doesn't recognize `python`, then you probably need to add Python to your PATH. Instructions for doing that are outside the scope of this tutorial.)

5. Go to `http://localhost:8000/` in your address bar. This address lets you view the local web server.

By default, web servers default to the `index.html` file in the directory, so it will show the Swagger UI file automatically. If the browser doesn't direct to `index.html`, add it manually: `http://localhost:8000/index.html`.

To stop the server, press **Ctrl+C** in your command prompt. If you closed your Command Prompt before stopping the service, type `ps`, find the process ID, then type `kill -9 <process ID>`

Mac: Run the Python simple HTTP server

1. In your terminal, browse to the Swagger UI `dist` directory.

To browse in your terminal, type `cd <folder name>` to move into the folder. Type `cd ..` to move up a directory. Type `ls` to see a list of the current directory's contents. If you're having trouble locating the `dist` directory in the command prompt, try this trick: type `cd`, press the spacebar, and then drag the `dist` folder directly into the command prompt. The path will be printed automatically.

2. Since Mac already has Python, you can just run the following in your terminal to launch simple server:

```
python -m http.server
```

If this command doesn't work, try it with the 3 in case you already have Python3 installed:

```
python3 -m http.server
```

The server starts:

```
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

3. Go to `http://localhost:8000/` in your address bar. This address lets you view the local web server.

By default, web servers default to the `index.html` file in the directory, so it will show the Swagger UI file automatically. If the browser doesn't direct to `index.html`, add it manually: `http://localhost:8000/index.html`.

To stop the server, press **Ctrl+C** in your command prompt. If you closed your Command Prompt before stopping the service, type `ps`, find the process ID, then type `kill -9 <process ID>`

For more details on using the Python simple server, see:

How do you set up a local testing server?

Learn about the [simple solution](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/set_up_a_local_testing_server) used in this guide at https://developer.mozilla.org/en-US/docs/Learn/Common_questions/set_up_a_local_testing_server for more details.

Customize the OpenAPI file

By default, SwaggerUI has the Petstore OpenAPI document configured in the url parameter in the `index.html` file. You need to swap in your local file instead.

1. Download the following OpenAPI document (right-click the link and save the YAML file to your desktop.):

[OpenWeather OpenAPI specification](https://idratherbewing.com/learnapidoc/docs/openapi_spec_and_generated_ref_docs/openapi_openweathermap.yml)

(https://idratherbewing.com/learnapidoc/docs/openapi_spec_and_generated_ref_docs/openapi_openweathermap.yml)

2. Drag your OpenAPI specification file, `openapi_openweathermap.yml`, into the `dist` folder. Your file structure should look as follows:

```
├── dist
│   ├── favicon-16x16.png
│   ├── favicon-32x32.png
│   ├── index.html
│   ├── oauth2-redirect.html
│   ├── swagger-ui-bundle.js
│   ├── swagger-ui-bundle.js.map
│   ├── swagger-ui-standalone-preset.js
│   ├── swagger-ui-standalone-preset.js.map
│   ├── swagger-ui.css
│   ├── swagger-ui.css.map
│   ├── swagger-ui.js
│   ├── swagger-ui.js.map
│   ├── swagger30.yml
│   └── openapi_openweathermap.yml
```

3. Inside your `dist` folder, open `index.html` in a text editor such as Visual Studio Code.

4. Look for the following code:

```
url: "http://petstore.swagger.io/v2/swagger.json",
```

5. Change the url value from `http://petstore.swagger.io/v2/swagger.json` to a relative path to your YAML file, and then save the file. For example:

```
url: "openapi_openweathermap.yml",
```

6. View the `index.html` file locally in your browser using the Python simple server. For example, go to `http://0.0.0.0:8000/` or `http://0.0.0.0:8000/index.html`.

If you go to the file path, such as `file:///Users/tomjoht/Downloads/dist/index.html`, you'll see a message that says "Failed to load API definition" note in the JavaScript Console that says "URL scheme must be "http" or "https" for CORS request." The SimpleServer simulates this http or https.

7. To stop the Python simpler server, press **Ctrl+C** in your terminal or command prompt.

3 Git repository

An important tool in the development and testing of any project is making use of a (distributed) version control software. The most widely used is Git based.

Create a private repository for the program used in the previous section and check if you are familiar with the main concepts related to using Git, like:

- Clone, commit, push ... operations
- Branches and pull requests
- Creating releases
- Integrating basic test operations as part of a Continuous Integration (CI) approach.

You can find these topics explained in:

- <https://www.atlassian.com/git>
- <https://git-scm.com/docs/gittutorial>
- <https://www.w3schools.com/git/>
- ... and in many other locations.