

Swagger Editor 3.0 Docker Container

- [Swagger Editor 3.0 Docker Container](#)
 - [Introduction](#)
 - [Assumptions](#)
 - [Create the Docker Container](#)
 - [Create Host directory which to mount in the container](#)
 - [Notes on the Container - **Read Before Opening**](#)
 - [Synchronisation of changes between Host and Container](#)
 - [What port does Swagger Editor listen on?](#)
 - [Create the example openapi.yaml API Specification \(OpenAPI 3.0.1\)](#)
 - [Use the container](#)
 - [Start the container](#)
 - [Connect to the running container](#)
 - [Test Swagger Editor on Host](#)
 - [Next Steps](#)
 - [Licensing](#)

Introduction

The intent of this document is to provide information on how to create a self-contained Docker container for API-First development using the [mwczapski/swagger-editor:1.0.0](#) image hosted on Docker Hub.

The container provides the means to:

- Run the Swagger Editor Server
- Convert YAML specification documents to JSON and the vice versa

[\[Top\]](#)

Assumptions

It is assumed in the text that Windows 10 with Debian WSL is the Host operating environment.

Make such changes, to the small number of commands that this affects, as you need to make it work in a regular Linux environment.

[\[Top\]](#)

Create the Docker Container

In this section a docker container with all that that is necessary to serve the Swagger Editor UI to the Host's Web Browser and convert between YAML and JSON openapi specifications will be created and started.

[\[Top\]](#)

Create Host directory which to mount in the container

Adjust Host paths above directory named **api** as you see fit.

```
HOST_DIR=/mnt/d/github_materials

mkdir -pv ${HOST_DIR}/swagger_editor/api
cd ${HOST_DIR}/swagger_editor
```

[\[Top\]](#)

Notes on the Container - **Read Before Opening**

Synchronisation of changes between Host and Container

On startup, the container starts the Swagger Editor server. This happens when the container is first started and when it is re-started.

The Swagger Editor server's `index.html` has been rigged to serve the file `openapi.yaml` from its own directory (`/swagger_tools/swagger_editor`). If the file is not available in the `/api` directory the server will fall back to serving the `/swagger_tools/swagger_editor/openapi.yaml` specification that was created during image build.

The source-of-truth `openapi.yaml` file for the container is the `/api/openapi.yaml` file in the container. This file, and all other files in the `/api` directory, are monitored for changes and when changes are detected this files are automatically copied to the location from which the Swagger Editor server expects to serve the `openapi.yaml` file. As soon as the `openapi.yaml` file becomes available in the `/api` container directory it will be copied to the Swagger Editor directory and served.

The command that accomplished it is reproduced below for your information. It is automatically run so there is no need to do anything to start the server and to restart the server when files in the container's `/api` directory change.

```
# nodemon -L -w index.html -w /api -x "cp -v /api/* /swagger_tools/swagger-editor/
&& http-server -p 3001"
```

The purpose of copying all files form the `/api` directory, and not just the `openapi.yaml` file, is to make it possible to embed in the `openapi.yaml description` properties hyperlinks to related documents in the same directory as the `openapi.yaml`, and have them available in the Swagger Editor in the usual (click on the link) way.

Please note that the Swagger Editor runs in the Web Browser in the Host environment.

The recommended container startup command, shown iin this document, mounts a host directory over the top of the container's `/api` directory. This gives the Swagger Editor the ability to write to the `/api` directory in the container and have the changes visible in the mapped host's directory. This gives external tools, like for example VSCode or IntelliJ running on the Host, the ability to edit the files in the Host's directory and have them "immediately" available to the Swagger Editor server in the container, and consequently to the Swagger Editor UI in the Web Browser as soon as the Web Browser page is refreshed.

[\[Top\]](#)

What port does Swagger Editor listen on?

Swagger Editor server inside the container listens on port **3001**. To change the port on which the Host listens, change the port mapping the container start command uses.

For example:

```
CONTAINER_MAPPED_PORTS=" -p 127.0.0.1:3210:3001/tcp "
```

will change the port the Hosts maps to the container's **3001** from **3001** to **3210**. The Host's web browser will need to use the url <http://localhost:3210/#> to connect to the Swagger Editor served from container.

[\[Top\]](#)

Create the example openapi.yaml API Specification (OpenAPI 3.0.1)

As already mentioned, the container expect the **openapi.yaml** file to be available in it's **/api** directory. As recommended, the container startup command will bind a Host directory to a **/api** directory in the container.

Let's create the **openapi.yaml** file in the bound host directory so that the container can access it.

```
HOST_DIR=/mnt/d/github_materials

cat <<- 'EOF' > ${HOST_DIR}/swagger_editor/api/openapi.yaml
openapi: "3.0.1"
info:
  title: Weather API
  description: |
    This API is a __test__ API for validation of local swagger editor
    and swagger ui deployment and configuration
  version: 1.0.0
servers:
  - url: 'http://localhost:3003/'
tags:
  - name: Weather
    description: Weather, and so on
paths:
  /weather:
    get:
      tags:
        - Weather
      description: |
        It is __Good__ to be a _King_
        And a *Queen*
        And a _Prince_
        And a __Princess__
        And all the Grandchildren
```

```

    And their Children
    operationId: getWeather
    responses:
      '200':
        description: 'All is _well_, but not quite'
        content: {}
      '500':
        description: Unexpected Error
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/response_500'
  components:
    schemas:
      response_500:
        type: object
        properties:
          message:
            type: string

```

EOF

[\[Top\]](#)

Use the container

Start the container

On the host, start the container with the following command, assuming

`/mnt/d/github_materials/swagger_editor/api` is the host directory to share.

Please note that the Windows version of Docker (which is what I use) wants a DOS'ish path

(`d:/github_materials/swagger_editor`) when run from the WSL Bash shell, rather than a WSL Linux'ish path, which would be something like `/mnt/d/github_materials/swagger_editor` in my Windows/WSL environment. If you Docker runs in the proper Linux/Unix environment the Host path would be a regular Linux/Unix path. Change as required.

Create the docker container start script:

```

HOST_DIR=/mnt/d/github_materials
cd ${HOST_DIR}

SOURCE_HOST_DIR=d:/github_materials/swagger_editor

HOST_LISTEN_PORT=3001
IMAGE_VERSION="1.0.0"
IMAGE_NAME="mwczański/swagger_editor"
CONTAINER_NAME="swagger_editor"
CONTAINER_HOSTNAME="swagger_editor"
CONTAINER_VOLUME_MAPPING=" -v ${SOURCE_HOST_DIR}/api:/api"

```

```
CONTAINER_MAPPED_PORTS=" -p 127.0.0.1:${HOST_LISTEN_PORT}:3001/tcp "
```

```
cat <<-EOF > start_swagger_editor_container.sh
```

```
docker.exe run \  
  --name ${CONTAINER_NAME} \  
  --hostname ${CONTAINER_HOSTNAME} \  
  ${CONTAINER_VOLUME_MAPPING} \  
  ${CONTAINER_MAPPED_PORTS} \  
  --detach \  
  --interactive \  
  --tty\  
  ${IMAGE_NAME}:${IMAGE_VERSION}
```

```
EOF
```

```
chmod u+x start_swagger_editor_container.sh
```

Start the container: `./start_swagger_editor_container.sh`

[\[Top\]](#)

Connect to the running container

The following command will connect us to the running container and offer the interactive bash shell to work in if required, such as for example when running the swagger-codegen to convert between yaml and json. I can think of no other reason it might be necessary work inside the container.

```
docker exec -it -w='/api' swagger_editor bash -l
```

[\[Top\]](#)

Test Swagger Editor on Host

The swagger-editor server is started when the container is created and started, and re-starts when the container is restarted.

With the container running, in a host web browser open the pre-configured API specification in the swagger-editor.

`http://localhost:3001/#`

Please note that the Swagger Editor is running in the Host's Web Browser and that it can import OpenAPI specifications from the Host and export / save OpenAPI specifications to the host, thus it can also be used as a local Swagger Editor on the Host.

Next Steps

Now that one can edit the API using a Swagger Editor hosted in the docker container it might be good to be able to generate API stubs to test the API.

This is coming in the next installment.

[\[Top\]](#)

Licensing

The MIT License (MIT)

Copyright © 2020 Michael Czapski

Rights to Docker (and related), Git (and related), Debian, its packages and libraries, and 3rd party packages and libraries, belong to their respective owners.

[\[Top\]](#)

2020/07 MCz