

能力目標202504

Swagger（openApi）用いてのSpringBootでのApi作成

■目次

1. Swaggerについて
2. 本ソースについて
3. ファイル構成について
4. 実施した作業手順（yamlファイルを手書きで記載した場合）
5. 本ソースベースでのApiの追加手順
6. yamlファイルGui作成とGui上でのコード自動生成

■Swaggerについて

Swaggerがなにかについては、下記ページなどを参考。

https://qiita.com/karasu_maru/items/342073fa7607fd4082bd

■本ソースについて

本ソースについては下記の内容を含んでいます。

1. 今回は**Swagger Spec**をyaml形式で手動作成し、openapi-generatorを使用しコードを自動作成を行い、実処理部分の実装後に動確を実施した作成物
2. Guiにて、Swaggerspec作成し自動生成を実施した際の出力結果物

■ファイル構成について

- SwaggerGuiSupplement配下
Guiでの自動生成に関する結果物を格納。
- openApiGeneratorCli配下
今回使用した「openapi-generator-cli」のjarファイルを格納
- 上記以外は、yamlファイルを手書きで動確を行った場合に作成したものとなります。

■実施した作業手順（yamlファイルを手書きで記載した場合）

1.OpenApi.yamlの作成

- ・ 自動生成するAPIの内容を記載する

yamlファイルを手動作成、または後述の「yamlファイルのGui作成」を参照。

2.codeGenerateExec.batの作成・実施（openApiGeneratorによるコード自動生成）

openapi-generator-cliのjarファイルを実行するbatファイル（codeGenerate.bat）を自動生成時のオプション

を引数で指定し実施。

※各batファイルについてはコンソールcommandを残すためbat化しているため手動でも実施可

■自動生成時のオプションについて

- ・ -i openapi.yaml（ファイル指定）
- ・ -g spring（今回はspringのためspring指定）
- ・ -o springApp（アプリケーション名）
- ・ --additional-properties=library=spring-boot,interfaceOnly=false

ライブラリにSpringBootを指定し、簡略化のためinterfaceOnlyはfalse

（trueはコントローラーのinterfaceのみ作成されるため、mainクラスなど自動生成されない）

※多言語や他の設定内容については、下記の公式Docを参照（springの参考元はspring.md）

<https://github.com/OpenAPITools/openapi-generator/tree/master/docs/generators>

3.APIコントローラーの作成（HelloApiController）

自動生成されたHelloAPIインターフェースを実装部分を作成。

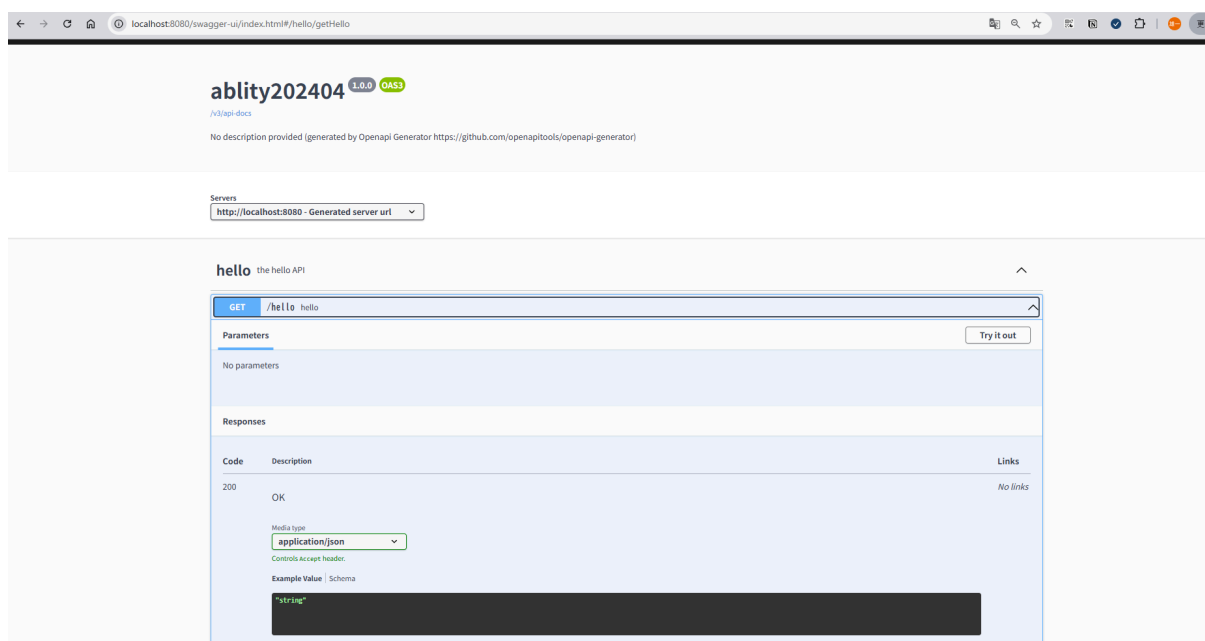
※今回は、簡略化のためOpenApi.yamlでGetのみとなり、処理も"Hello"を返すだけとする。

4.Docker関連ファイルの作成

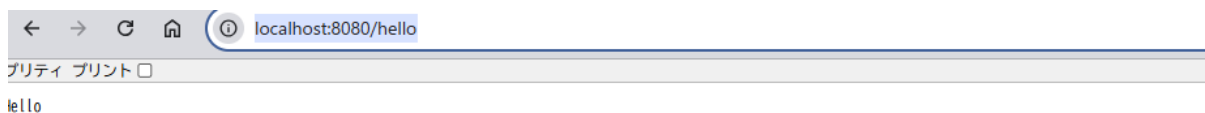
- Dockerfile作成（今回のテーマではないため）
- dockerBuild.bat dockerRun.bat作成（dokcerBuild,runのcommand保管用Bat）

5.作成物確認

- SwaggerUi（SwaggerでのAPI仕様書）「<http://localhost:8080/swagger-ui/index.html>」 接続



- Api呼び出し「<http://localhost:8080/hello>」 確認



■本ソースベースでのApiの追加手順

1.Cドライブ直下に「openApiGeneratorCli」フォルダをコピー

＊Cドライブ以外を使用したい場合は、任意のディレクトリにコピー後、

codeGenerate.batのファイルパスを変更してください。

2.OpenApi.yamlにAPI内容を追記

3.codeGenerateExec.batを実施

4. 追記したAPIのコントローラーなど実処理を作成。

5.dockerBuild.batを実施

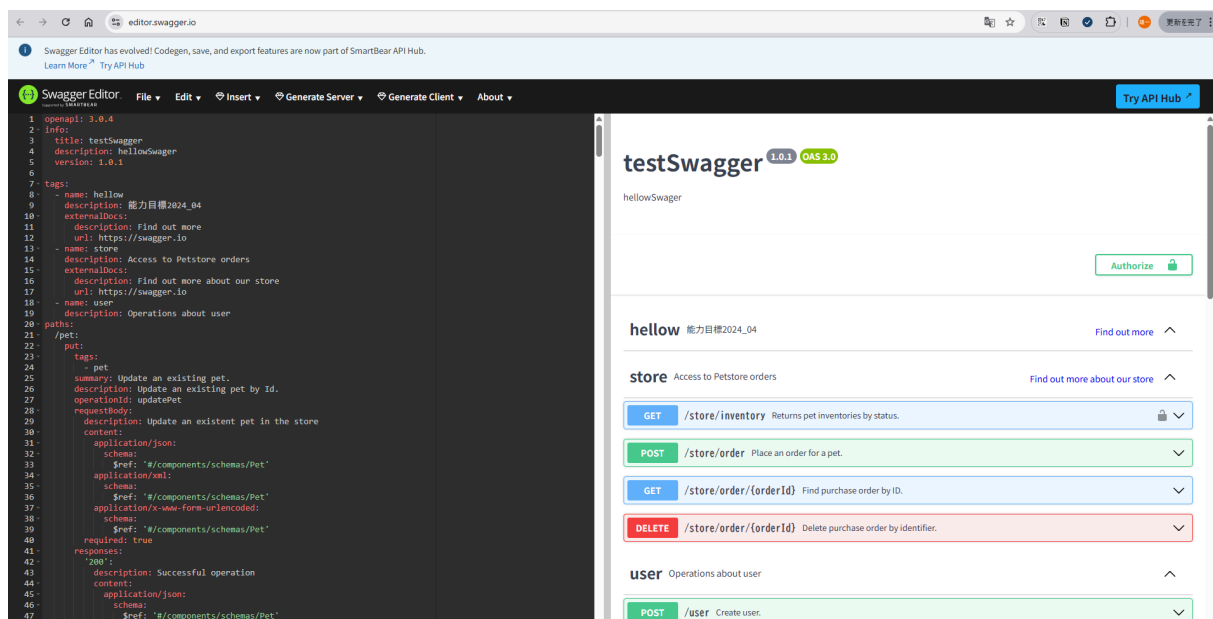
6.dockerRun.batを実施

7.接続確認

■yamlファイルGui作成とGui上でのコード自動生成

1.公式サイトでの作成「<https://editor.swagger.io/>」

※こちらでの作成でも構わないが、セキュリティ上問題がある（現場を想定した場合）

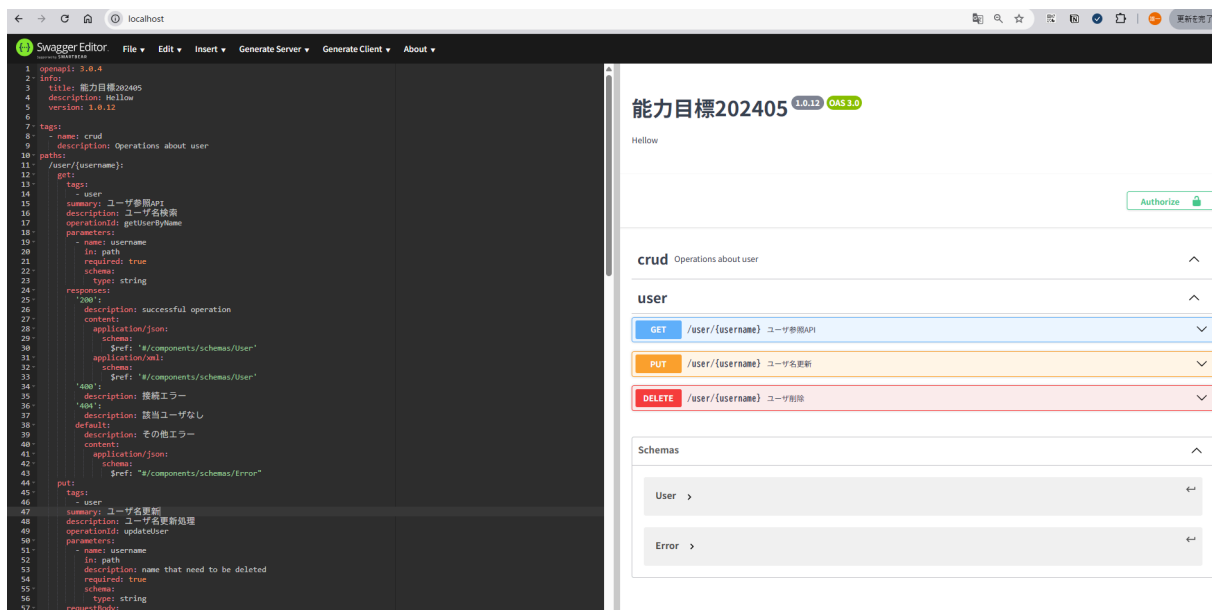


2.Dockerでimageを使用しCLI で直接起動後の作製。

A. swagger-editor,swagger-codegen-cliのimageを取得後、DockerRunで起動

※実施内容は「SwaggerGuiSupplement/SwaggerGuiSupplement.bat」を参照。

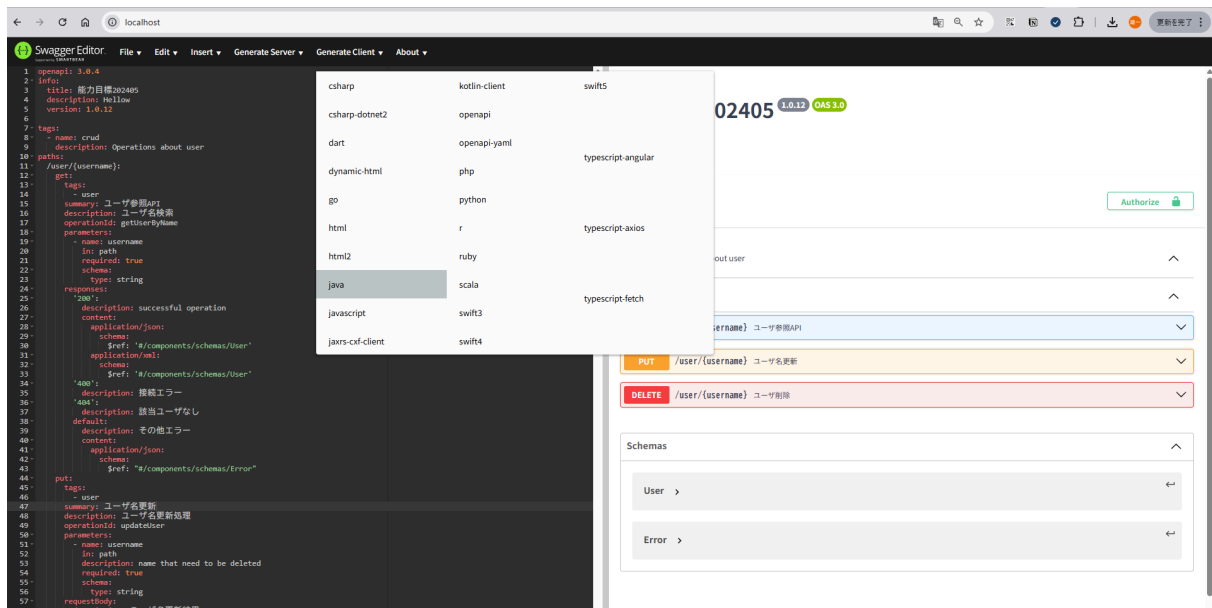
下記の通り、公式ブラウザの内容がローカル環境で実施が可能となる。



B.以下のボタンにより、作成したyamlファイルがダウンロードできる



C.Gui上でのコード自動生成



D.Gui上で作成したyamlファイルと自動生成されたファイルについては
SwaggerGuiSupplement