Favourite Recipes Documentation

I would like to thank you for giving me this challenge. It was pretty interesting for me, I hope I did a good job regards :)

Mohamed Ibrahim Ali

Let's get started

Main Architecture

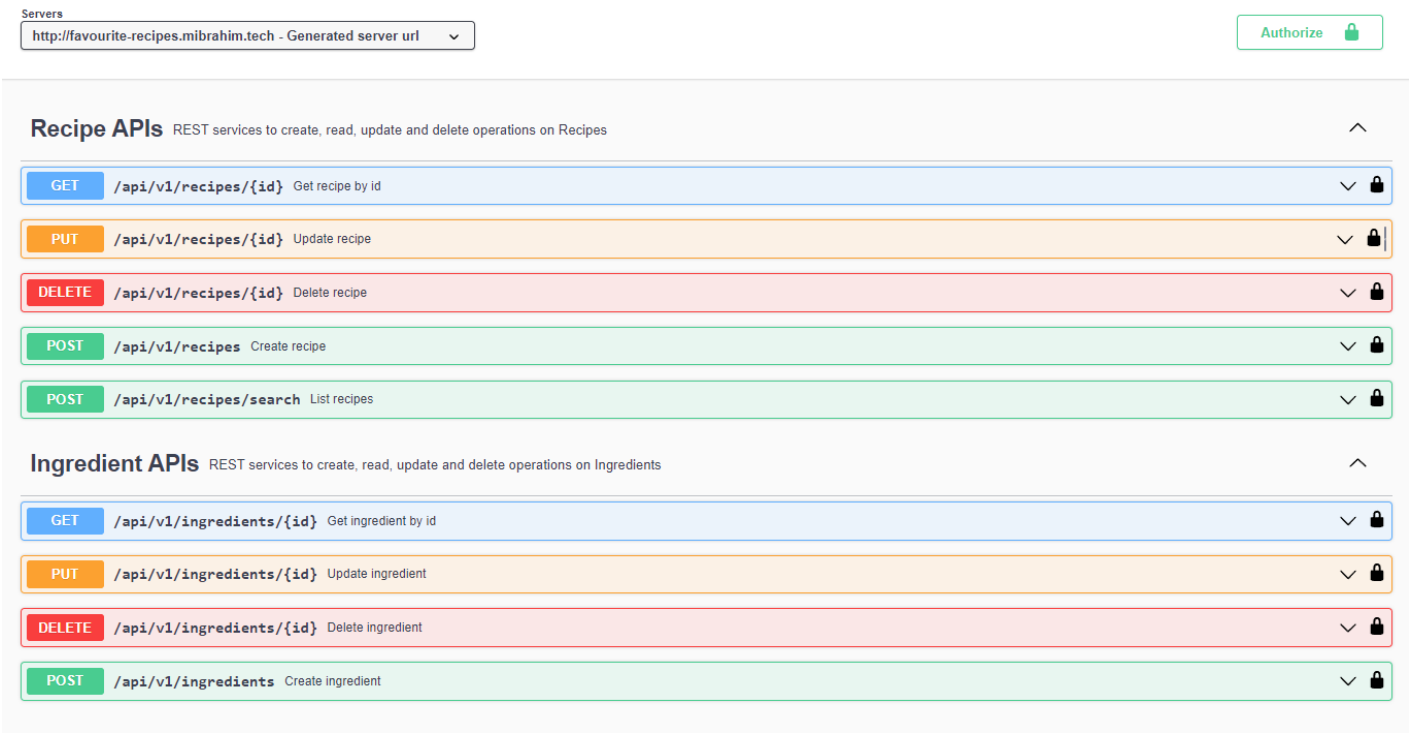
Design Points:

- I selected Spring Boot framework as it provides an easy and faster way to set up, auto configure, and run both simple and web-based applications, without relying on an external web server, by embedding a web server such as Tomcat or Netty into application during the initialization process and also easy to build, test and deploy on clusters and virtual machine through dockr for example.
- Open API to defines a standard, language-agnostic interface to RESTful and force any changes in the contract to all teams within the development stages continuslly, and Swagger as it simplify API development for users, teams, and enterprises with an open source and professional toolset and allow clients to test APIs easily throgh a fancy forums.
- Spring security with Auth0 to authenticate and authorize users who call the protected APIs, and we can take advantage of many features provided by Auth0 Single Sign On (SSO) in the future if we want.
- PostgresSQL is an open source relational database so it reduce costs and doesn't cost anything – no license fees!, driven by a big and great community which contrubutes for more than 25 years now, security and scalabilty.
- Flyway as a database migration tool.
- Jacoco for code coverage

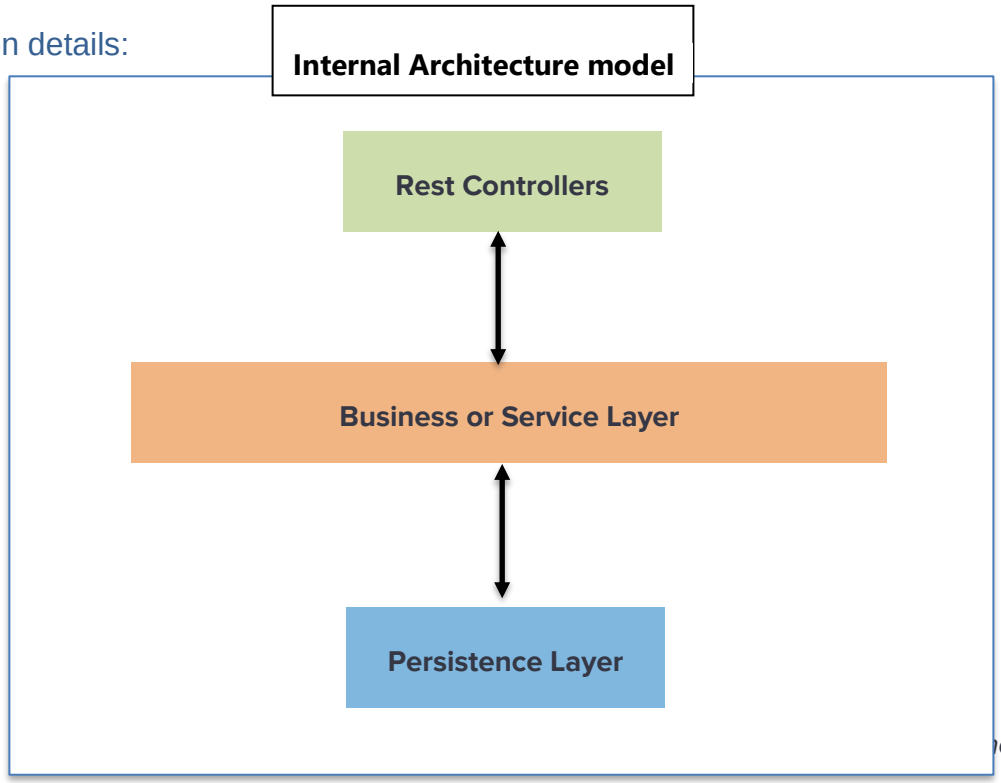
Favourite Recipes Documentation

- I assumed that users will create their recipes and ingredients in one request and same also for update, not through many requests like creating recipe first then add ingredients to it, so I went with the first approach of one request for create and one request for update.

Endpoints: (**I have been deplyed it to cloud so you can check it here from here** Swagger REST APIs)



Implementation details:



Requests comes to Rest controllers after authentication and authorization, regardless it bindly delegate the requests to business or service layer.

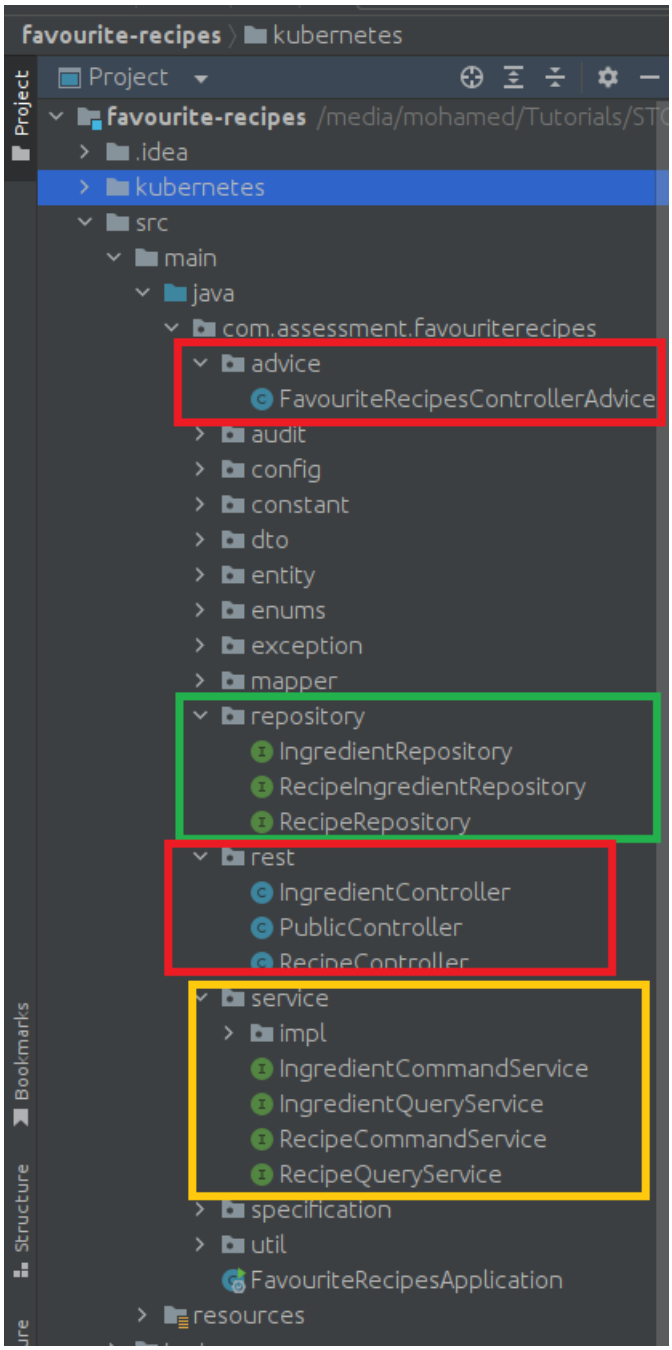
Business or service layer responsible for handling requests according by interaction with persistence layer in adition to processing the business logic.

FavouriteRecipesControllerAdvice acts as a global exception handler for rest controllers to handle custom FavouriteRecipeException and it maps each unhandled exception and map it to correct http status and http body like that:

404
Undocumented Error: response status is 404

Response body

```
{
  "code": "1001",
  "messageKey": "error.recipe.not.found",
  "value": "recipeId"
}
```



Code Coverage:

favourite-recipes

favourite-recipes

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.assessment.favouriterecipes.mapper		92%		64%	10	28	10	78	0	14	0	3
com.assessment.favouriterecipes.service.impl		100%		n/a	0	33	0	97	0	33	0	2
com.assessment.favouriterecipes.rest		100%		n/a	0	14	0	23	0	14	0	3
com.assessment.favouriterecipes.util		100%		100%	0	10	0	20	0	6	0	3
com.assessment.favouriterecipes.exception		100%		n/a	0	4	0	8	0	4	0	1
com.assessment.favouriterecipes.audit		100%		n/a	0	2	0	4	0	2	0	1
com.assessment.favouriterecipes		100%		n/a	0	1	0	1	0	1	0	1
Total	19 of 966	98%	10 of 36	72%	10	92	10	231	0	74	0	14

Build & Run:

Prerequisites:

1. Docker, Docker compose, Maven and JDK 11 installed and properly configured.
2. Ensure ports [8080, 5432] are free.

Steps to build and run local:

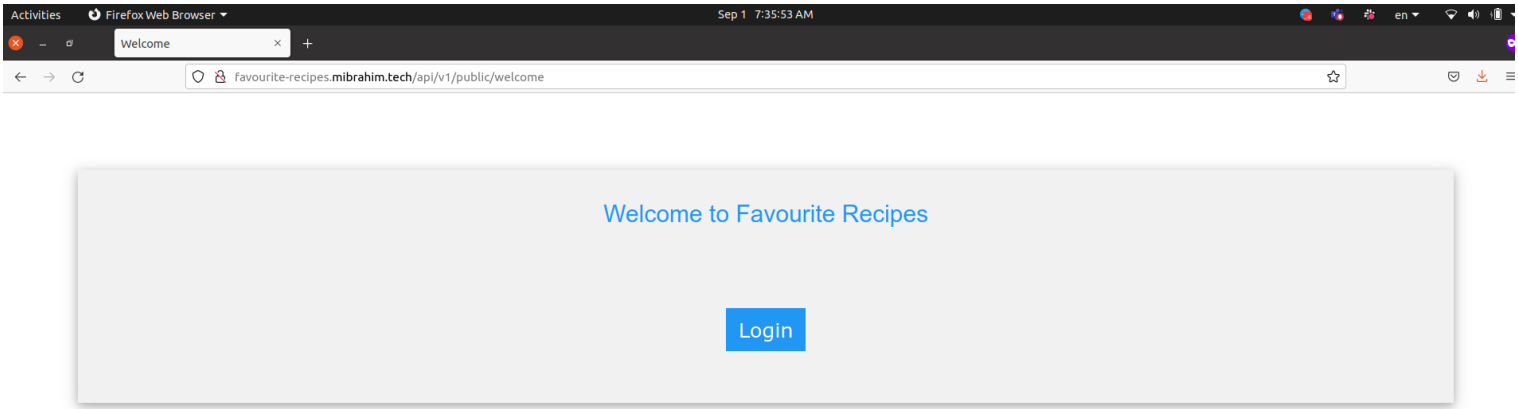
1- Clone the project into your machine from Github link https://github.com/mibrahim-iti/favourite-recipes.git
2- Open teminal.
3- Change directory (cd) to the favourite-recipes project folder.
4- Give execute permission to ***build-app-image-local.sh*** file (this is a bash script file which build the project docker image inside your local docker environment) by using next command **chmod 777 build-app-image-local.sh**
5- Type the next command in your terminal to build and run the application on your local docker **./build-app-image-local.sh && docker compose up**

Now you can go to welcome page from that link:
http://localhost:8080/api/v1/public/welcome

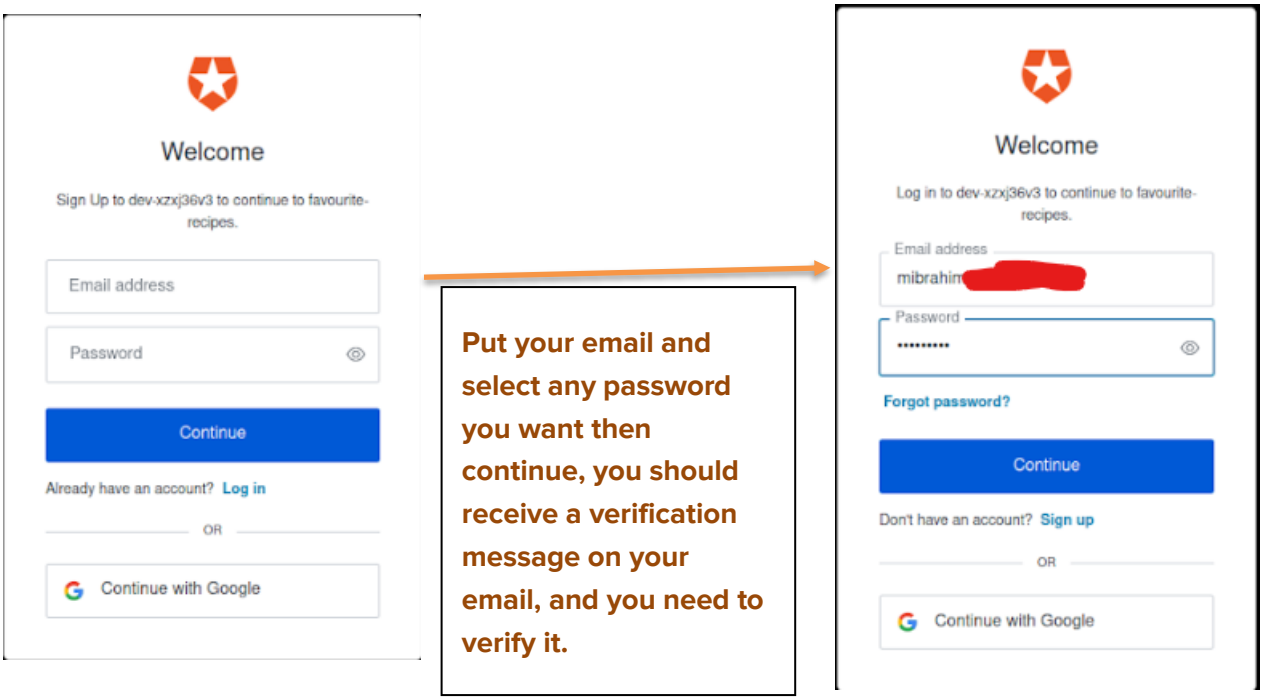
I already deployed it to cloud and I provided the link for swagger-ui previously, but here it is again Swagger REST APIs (Make sure you read next part of how to use the application so you see how to generate the acccess token and use it to authenticate your requests with swagger ui) but we can do this manually by following the next steps if you are using Linux.

Try and play with favourite-recipe application:

After you see the application is up and running so you can put the next url in the browser [**http://localhost:8080/api/v1/public/welcome**] which equivilant to what I deploye on the cloud in the next link [**http://favourite-recipe.mibrahim.tech/api/v1/public/welcome**] then you must see the next welcome login screen.



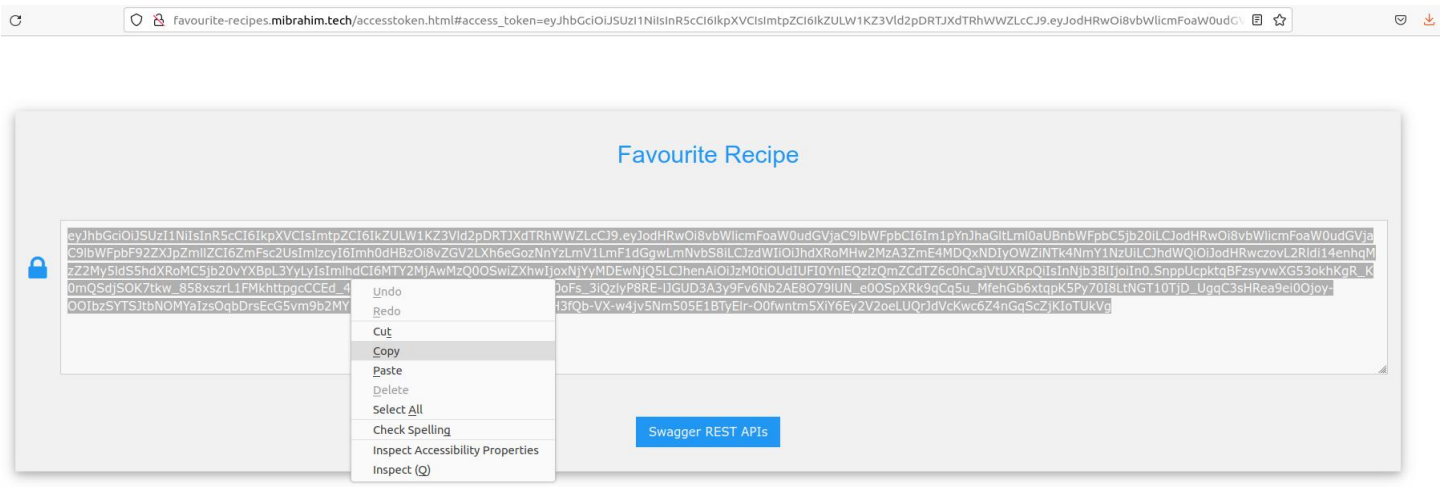
Click on Login and this will redirect you to **Auth0** service to login or sign up.



By: *Mohamed Ibrahim Ali*

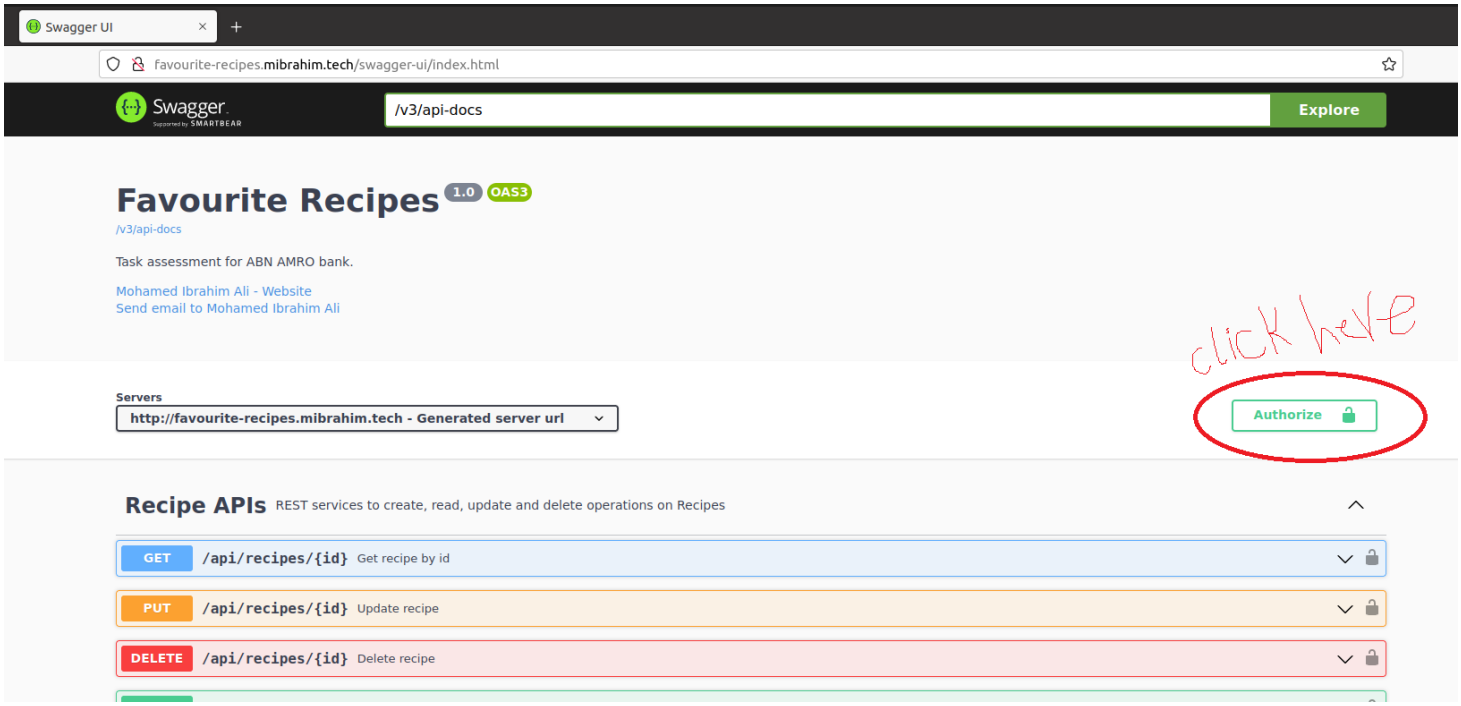
Favourite Recipes Documentation

By accept giving the application access permission, now you can login with your email and the pasword you created for the app.



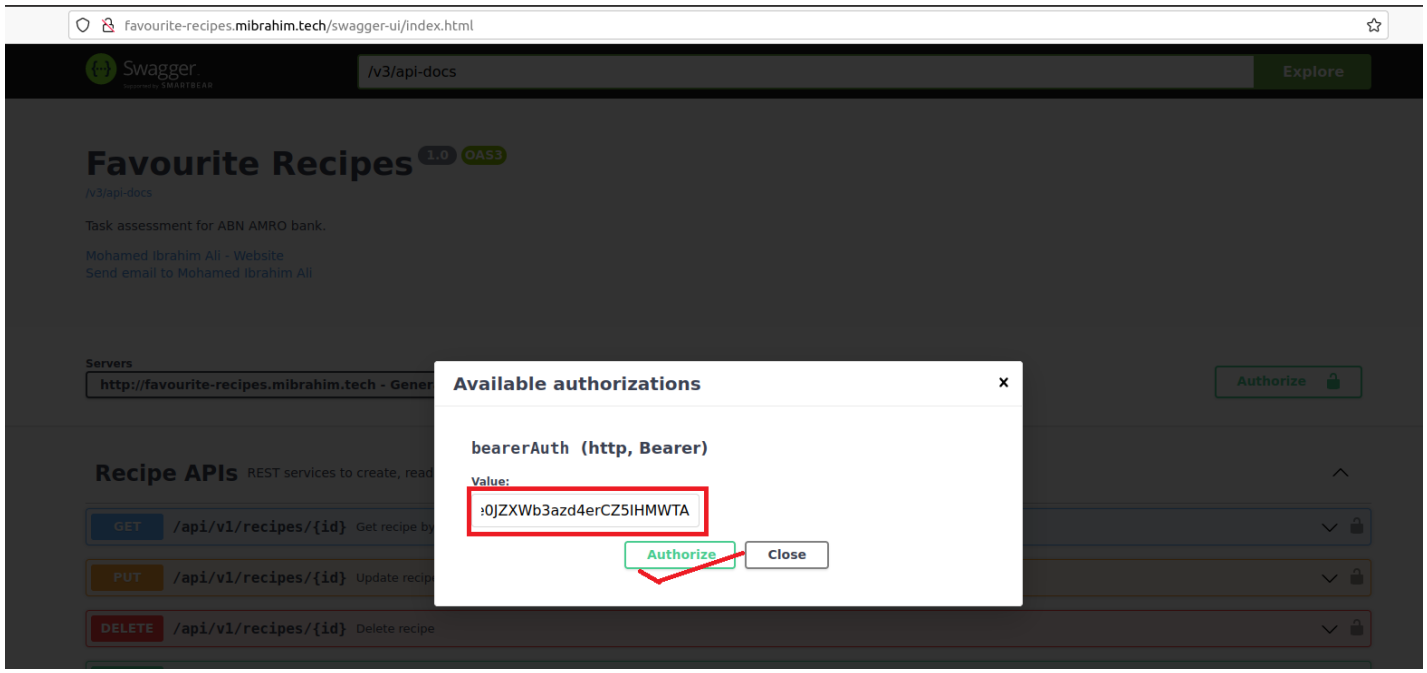
And after successfull login you will be redirect to **access token** page and you need to select the token and copy it to your clipboard so you can use it in swagger ui.

Click on on **Swagger REST APIs** you must see next page of Swagger:

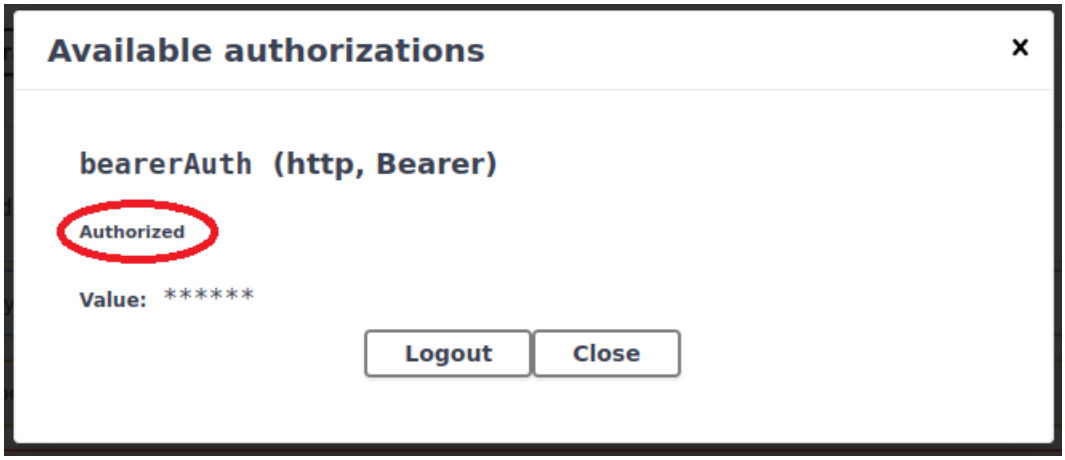


Put the token in the input text box and click Authorize

By: *Mohamed Ibrahim Ali*



You can see that you are **authorized** to use the REST services through Swagger UI so you test using it now.



Now you can start playing with application…

By: *Mohamed Ibrahim Ali*

Favourite Recipes Documentation

Swagger UI

favourite-recipes.mibrahim.tech/swagger-ui/index.html#/Ingredient APIs/createIngredient

Request body required

application/json

```
{
  "name": "Mohamed Ibrahim Ingredient"
}
```

Execute | Clear

Responses

Curl

```
curl -X 'POST' \
  'http://favourite-recipes.mibrahim.tech/api/v1/ingredients' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IkZULW1KZ3Vld2pDRTJXdTRhWWZLcCJ9.eyJodHRwOi8vbWlicmFoaW0udGVjaC9lbWFpbCI6ImlpYnJhaGltLml0aUBnbWFpbC5jb20iLCJodHRwOi8vb
  -H 'Content-Type: application/json' \
  -d '{
  "name": "Mohamed Ibrahim Ingredient"
}'
```

Request URL

```
http://favourite-recipes.mibrahim.tech/api/v1/ingredients
```

Server response

Code	Details
201 *Undocumented*	**Response body**

```
1
```

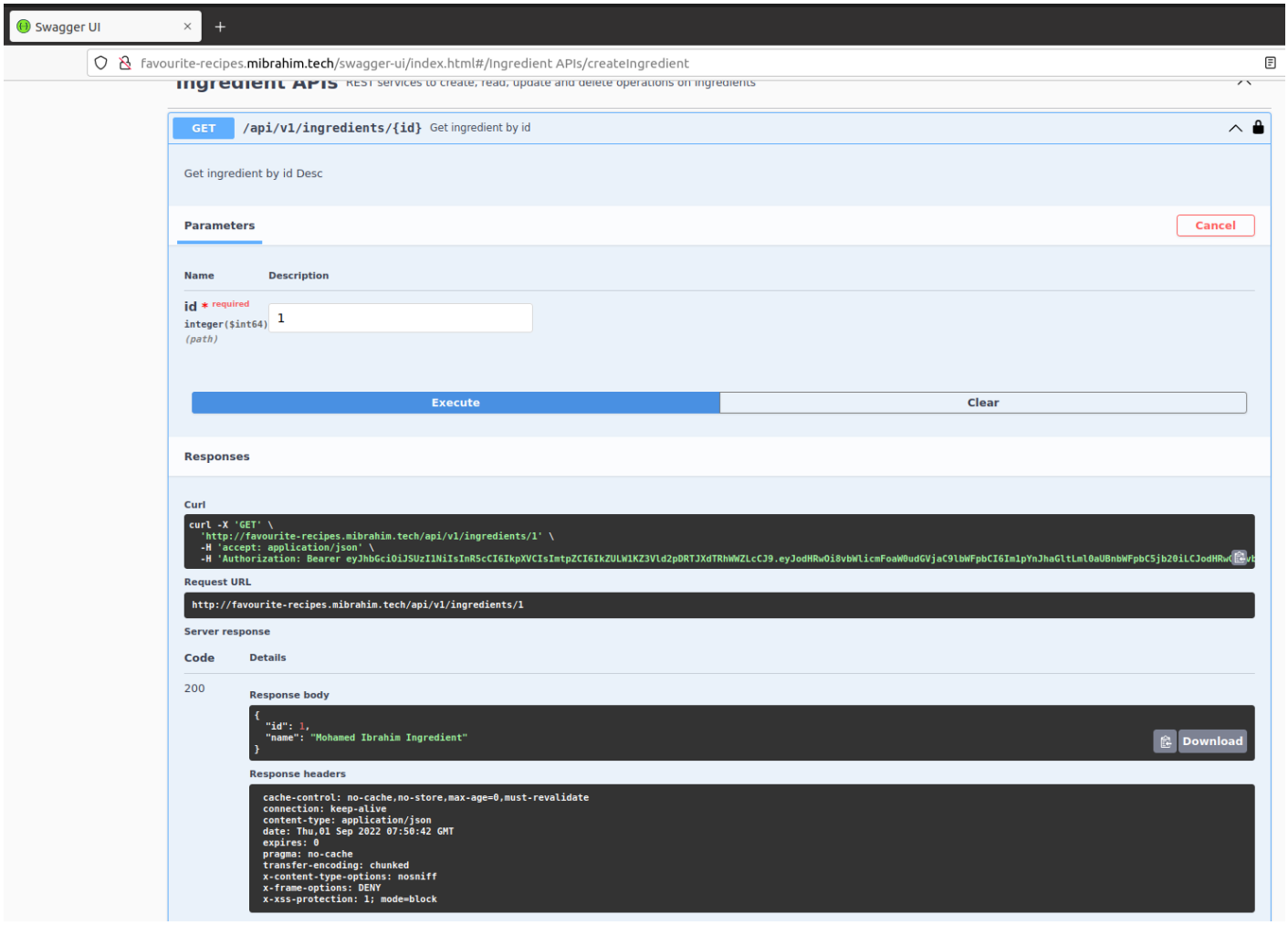
Download

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/json
date: Thu,01 Sep 2022 07:50:16 GMT
expires: 0
pragma: no-cache
```

By: *Mohamed Ibrahim Ali*

Favourite Recipes Documentation



Finally I would like to thank you again for such interesting task.
If you have any question or something is not clear enough
You are more than welcome to contact me.

mibrahim.iti@gmail.com
https://www.linkedin.com/in/mohamed-ibrahim-ali/
https://github.com/mibrahim-iti/favourite-recipes

By: *Mohamed Ibrahim Ali*