What’s Cooking Application Guide

Running Application

There are a few ways for us to run this program. Below we will show two ways of doing it. One is using the IntelliJ IDE, which is used to code the application, and another one is using the GIT Bash commands.

Using IntelliJ IDE

1. Go to the folder of the application following the path: src\main\java\com\swe\whatcooking
2. Run the WhatsCookingApplication.java file

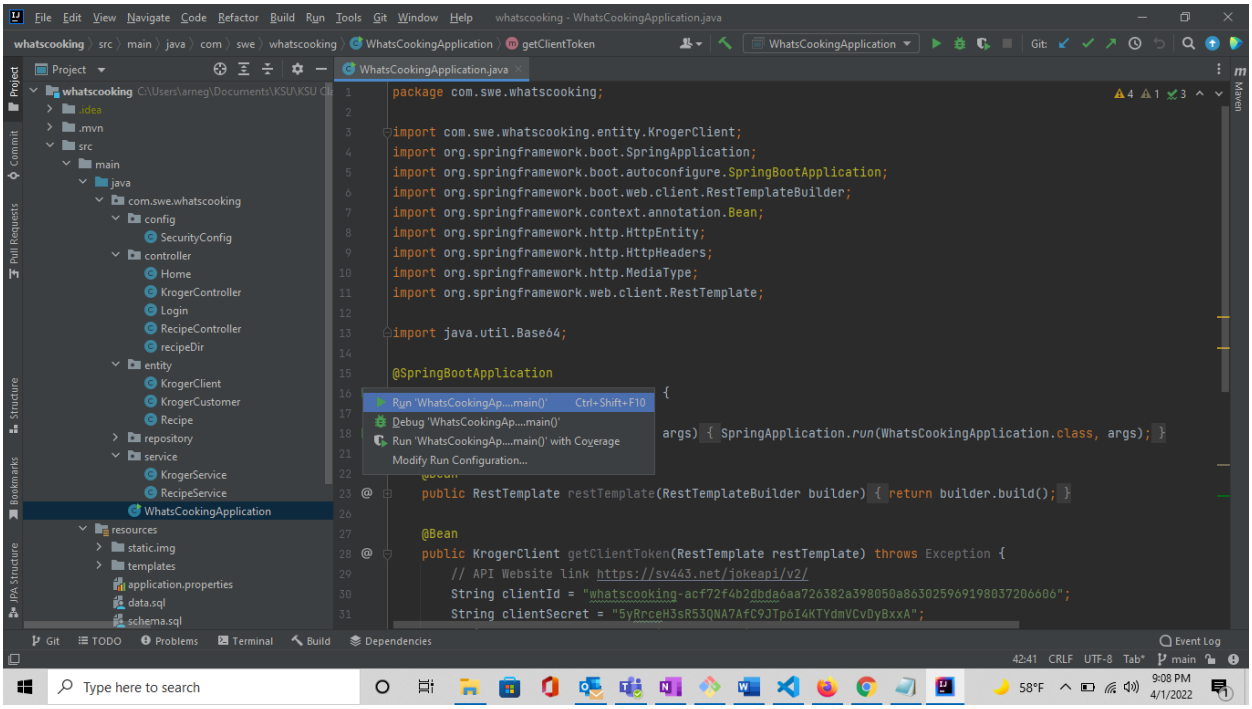


Figure 1- IntelliJ IDE View

Using Git Bash

The steps below include the cloning of the project. If you have done the cloning already, go to the main folder in which the POM file will be available along with the read me and such, and enter only the third command as shown in figure 2.

```
1. $ git clone https://github.com/marcusdorsey1/WhatsApp2.git
2. $ cd WhatsApp2
3. $ ./mvnw spring-boot:run
```

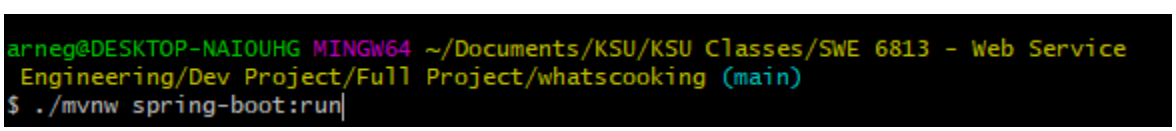
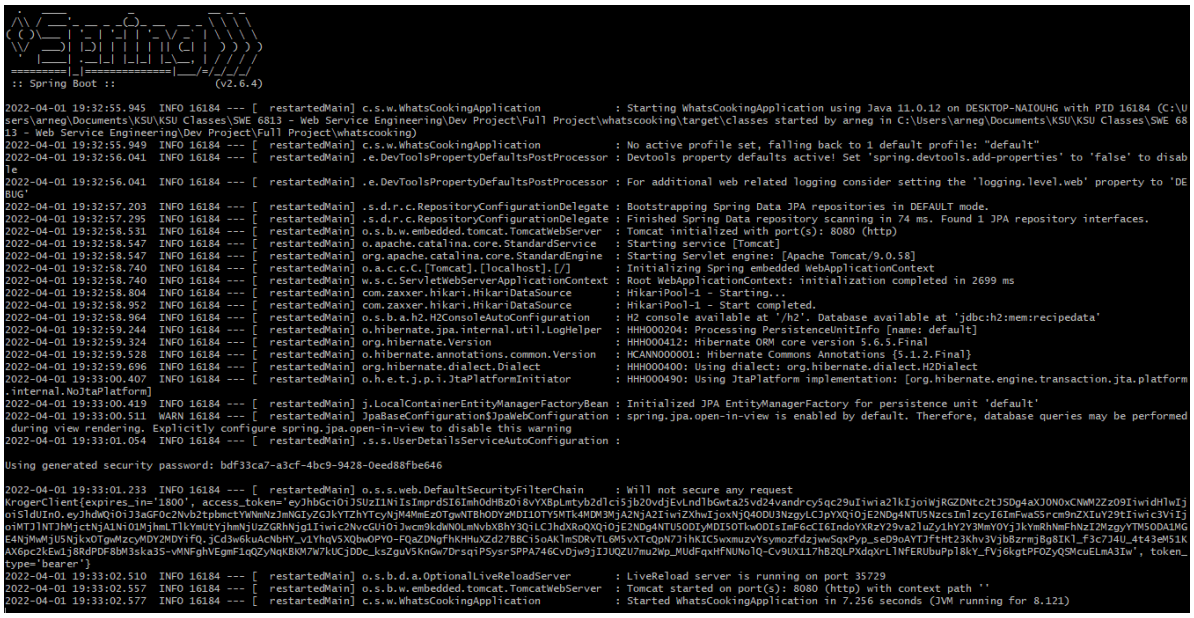


Figure 2- Running Spring Boot Application Command



```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.4)

2022-04-01 19:32:55.945  INFO 16184 --- [  restartedMain] c.s.w.WhatsCookingApplication            : Starting WhatsCookingApplication using Java 11.0.12 on DESKTOP-NAIOUHG with PID 16184 (C:\U
sers\arneg\Documents\KSU\KSU Classes\SWE 6813 - Web Service Engineering\Dev Project\Full Project\whatscooking\target\classes started by arneg in C:\Users\arneg\Documents\KSU\KSU Classes\SWE 68
13 - Web Service Engineering\Dev Project\Full Project\whatscooking)
2022-04-01 19:32:55.949  INFO 16184 --- [  restartedMain] c.s.w.WhatsCookingApplication            : No active profile set, falling back to 1 default profile: "default"
2022-04-01 19:32:56.041  INFO 16184 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disab
le
2022-04-01 19:32:56.041  INFO 16184 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DE
BUG'
2022-04-01 19:32:57.203  INFO 16184 --- [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-04-01 19:32:57.295  INFO 16184 --- [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 74 ms. Found 1 JPA repository interfaces.
2022-04-01 19:32:58.531  INFO 16184 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2022-04-01 19:32:58.547  INFO 16184 --- [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2022-04-01 19:32:58.547  INFO 16184 --- [  restartedMain] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.58]
2022-04-01 19:32:58.740  INFO 16184 --- [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2022-04-01 19:32:58.740  INFO 16184 --- [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2699 ms
2022-04-01 19:32:58.804  INFO 16184 --- [  restartedMain] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2022-04-01 19:32:58.952  INFO 16184 --- [  restartedMain] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2022-04-01 19:32:58.964  INFO 16184 --- [  restartedMain] o.s.b.a.h2.H2ConsoleAutoConfiguration    : H2 console available at '/h2'. Database available at 'jdbc:h2:mem:recipedata'
2022-04-01 19:32:59.244  INFO 16184 --- [  restartedMain] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2022-04-01 19:32:59.324  INFO 16184 --- [  restartedMain] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.6.5.Final
2022-04-01 19:32:59.528  INFO 16184 --- [  restartedMain] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2022-04-01 19:32:59.696  INFO 16184 --- [  restartedMain] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
2022-04-01 19:33:00.407  INFO 16184 --- [  restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform
.internal.NoJtaPlatform]
2022-04-01 19:33:00.419  INFO 16184 --- [  restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2022-04-01 19:33:00.511  WARN 16184 --- [  restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed
 during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2022-04-01 19:33:01.054  INFO 16184 --- [  restartedMain] .s.s.UserDetailsServiceAutoConfiguration :

Using generated security password: bdf33ca7-a3cf-4bc9-9428-0eed88fbe646

2022-04-01 19:33:01.233  INFO 16184 --- [  restartedMain] o.s.s.web.DefaultSecurityFilterChain     : Will not secure any request
KrogerClient{expires_in='1800', access_token='eyJhbGciOiJSUzI1NiIsImprdSI6Imh0dHBzOi8vYXBpLmtyb2dlci5jb2OvdjEvLndlbGwta25vd24vandrcy5qc29uIiwia2lkIjoiWjRGZDNtc2tJSDg4aXJONOxCNWM2Zz09IiwidHlwIj
oiSldUIn0.eyJhdWQiOiJ3aGFOc2Nvb2tpbmctYWNmNzJmNGIyZGJkYTZhYTcyNjM4MmEzOTgwNTBhODYzMDI1OTY5MTk4MDM3MjA2NjA2IiwiZXhwIjoxNjQ4ODU3NzgyLCJpYXQiOjE2NDg4NTU5NzcsImlzcyI6ImFwaS5rcm9nZXIuY29tIiwic3ViIj
oiMTJlNTJhMjctNjA1Ni01MjhmLTlkYmUtYjhmNjUzZGRhNjg1IiwiczZnRGUiOiJwcm9kdWNOLmNvbXBhY3QiLCJhdXRoQXQiOjE2NDg4NTU5ODIyMDI5OTkwODIsImF6cCI6IndoYXRzY29va2luZy1hY2Y3MmYOYjJkYmRhNmFhNzI2MzgyYTM5ODA1MG
E4NjMwMjU5NjkxOTgwMzcyMDY2MDYifQ.jCd3w6kuAcNbHY_vlYhqVSXQbwOPYO-FQaZDNgfhKHHuXZd27BBCi5oAKlmSDRvTL6MSvXTcQpN7JihKIC5wxmuzvYsymozfdzjwwSqxPyp_seD9oAYTJftHt23Khv3Vjb8zrmj8g8IKl_f3c734U_4t43eM5iK
AX6pc2kEwlj8RdPDF8bM3ska35-vMNFghVEgmF1qQZyNqKBKM7W7kUCjDDc_ksZguV5KnGw7DrsqiPSysrSPPA746CvDjw9jIJUQZU7mu2Wp_MUdFqxHfNUNolQ-Cv9UX117hB2QLPXdqXrLlNfERUbuPpl8kY_fVj6kgtPFOZyQSMcuELmA3Iw', token_
type='bearer'}
2022-04-01 19:33:02.510  INFO 16184 --- [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2022-04-01 19:33:02.557  INFO 16184 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2022-04-01 19:33:02.577  INFO 16184 --- [  restartedMain] c.s.w.WhatsCookingApplication            : Started WhatsCookingApplication in 7.256 seconds (JVM running for 8.121)
```

Figure 3- Spring Boot Application Running Log

Test Features

Our application will get started in localhost port 8080 the machine. The in order to start using the application, we will be using the URLs listed in each of the features to view it's progress. Navigate to localhost:8080/home to start the testing of the features. You will see all functionalities in order for best experience.

User Authentication

The application is password protected. In order to log in, we will need to enter the credentials as follows. username = **user** and password = password. We can copy that password and enter it on the password section of the form as shown in figure 5 and press the Sign in button to log into the application.

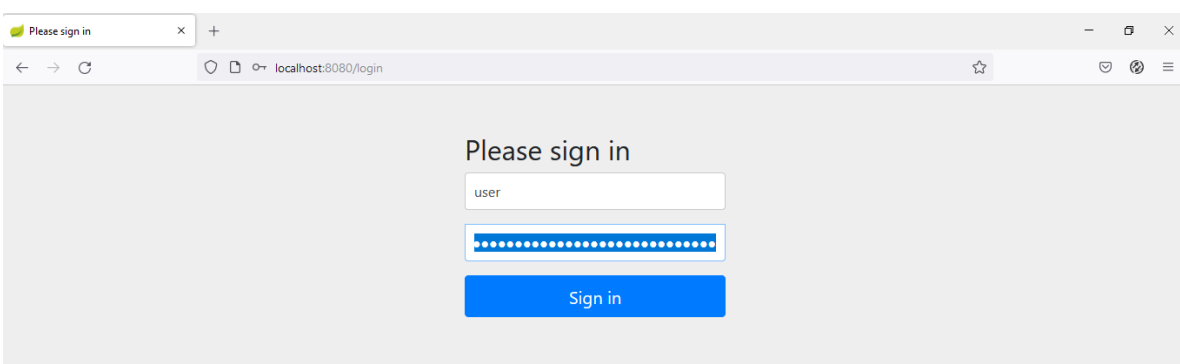


Figure 4- User Log In Page

Dynamic Web Application

You will be taken to the home page of our application. Currently this shows a dynamic view of the layout we would like to show our users. From this page we can select the Details button to see more information about a particular recipe. Press any Details button under the recipes to test our next Feature.

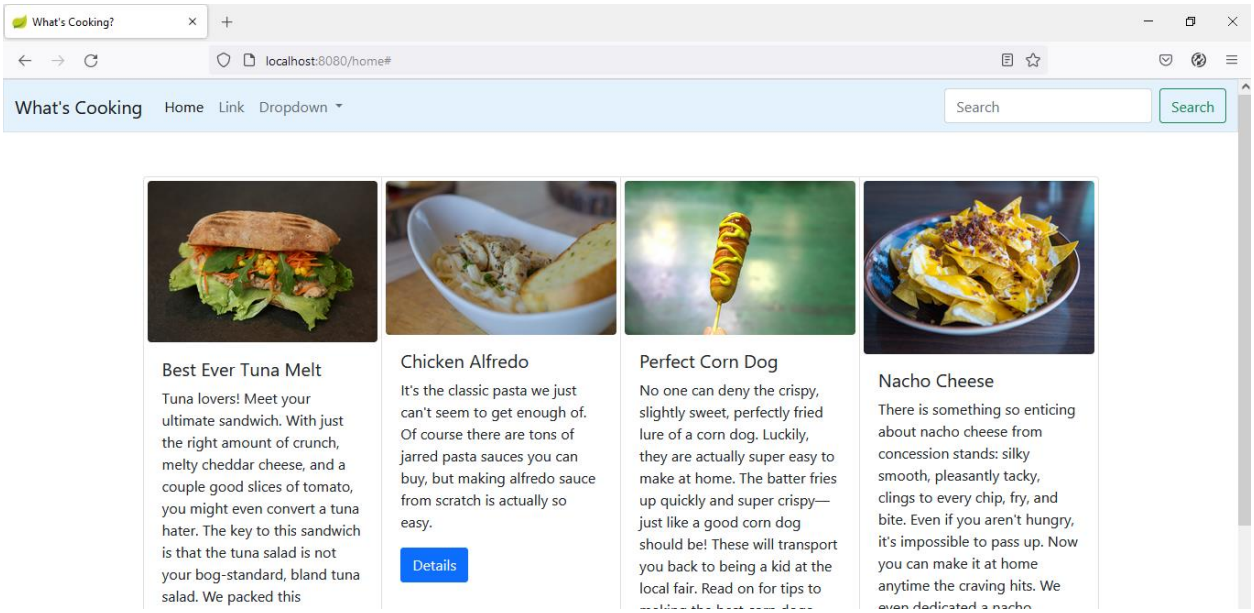


Figure 5- What's Cooking Home Page

Kroger API Functionality

In order to test this functionality, the user must have a Kroger Account or can create a new one while authorizing our application to add items to the cart. To test this feature, we must go to our home page and enter into the details of the Chicken Alfredo recipe, as it is the one hardcoded to use the functionality. Once you are in the details view as shown in figure 7, we can press the button Order Ingredients to start the process of adding items to the Kroger cart.

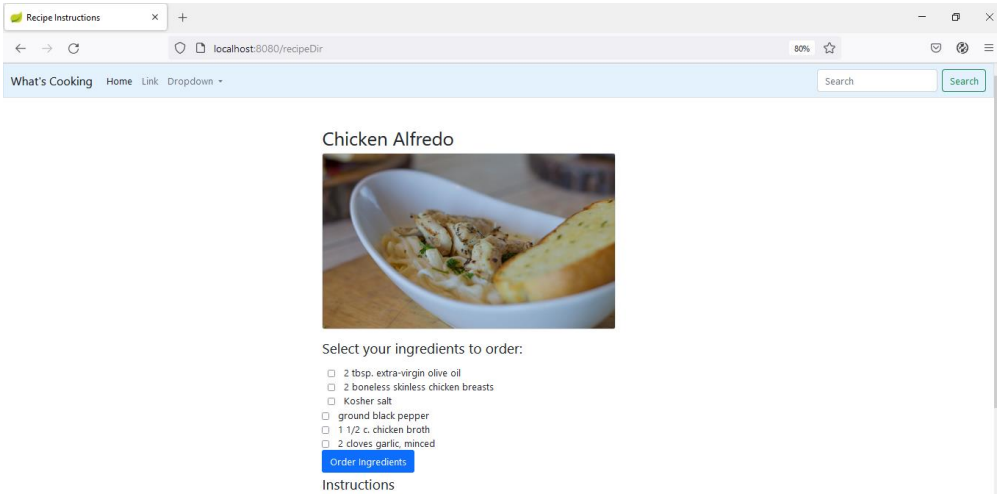


Figure 6- Recipe Details View

Kroger Cart Authorization

Once the button to Order Ingredients has been pressed, the application will take us to the Kroger Authorization as shown in figure 8. Enter a personal Kroger Login information or Sign Up for an account if we do not have one. After agreeing to the terms, the application will take us back to our home page. When we go back to the Kroger website link and log into the account, we will see the items there. This functionality is currently hard coded into the system to prove those items.

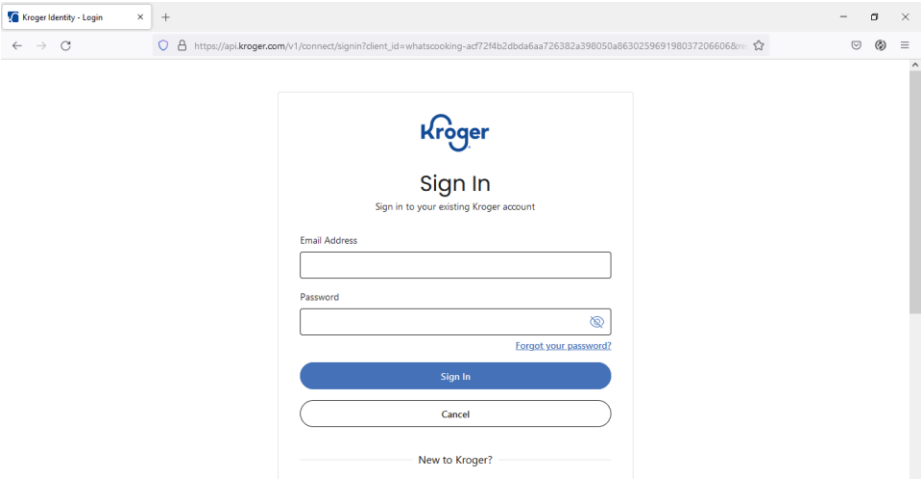


Figure 7- Kroger Authorization Page

API Endpoints

Finally, our last feature provides a small REST API endpoint in which we will share the recipes entered in our system to the world. In order to see all the API endpoints, we will have to go to http://localhost:8080/swagger-ui/index.html URL. This endpoint will list all the API endpoints available in our application. You can use this to execute the commands and see the response. Currently it uses Basic authentication which is the same username and password as the website.

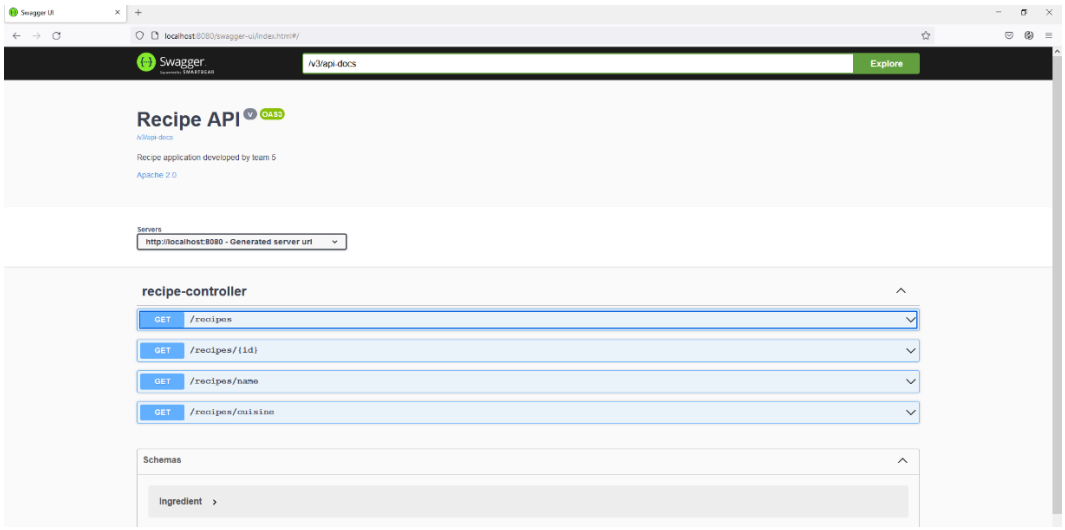


Figure 8- Recipe API Documentation