Prerequisites: Docker, Java17(Optional), maven(Optional), docker-compose, Postman(Optional)

Optional prerequisites are needed if you want to build the code. To run the code, you simply need docker and docker-compose installed on your system.

#Please install the prerequisites before running the below commands

#If you want to build images locally, these are the steps.

mvn clean install -DskipTests=true

./run.sh

#Alternatively you can download the images from docker hub and get the application up and running

docker-compose up

#Wait for some time aproximate 15 minutes after running this command until you can see keycloak running on localhost:8080

#If you are using Mac, open terminal and type the command:

sudo nano /etc/hosts

Add “127.0.0.1 keycloak” as shown in the image

A screenshot of a computer

Description automatically generated

#If you are using Windows, open terminal and type the command:

c:\Windows\System32\Drivers\etc\hosts

Add 127.0.0.1 ‘tab’ keycloak as shown in the above image

#Open your browser, type

Localhost:8080

#You must see keycloak running

A screenshot of a computer

Description automatically generated

#Go to Administration Console

#Sign in using username as admin and password as admin

A screenshot of a computer

Description automatically generated with medium confidence

# In the Clients tab select spring-cloud-client (the last client ID shown in the picture)

A screenshot of a computer

Description automatically generated with medium confidence

#In the Credentials tab click on Regenerate Secret.

#Copy the generated key

#There are two approaches to execute requests on the application.

1. Using Postman
2. Using Curl

If you do not want to use Postman, move to the bottom part of this file showing the way to execute requests using Curl

**Using Postman**

#Open Postman

#On the top left near Scratch Pad there is a button called Import.

# Click on Import

Import all the three files given,

Order Service.postman\_collection.json, Product Service.postman\_collection.json, Inventory Service.postman\_collection.json

#Now click on Authorization

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

#From the dropdown in the type select OAuth 2.0

#On the right to the type in the text box with the label Client Secret, paste the key generated using KeyCloak

A screenshot of a computer

Description automatically generated with medium confidence

#Now scroll down and select the orange button labeled ‘Get New Access Token’

#Click on Proceed

#Click on Use token

A screenshot of a computer

Description automatically generated with medium confidence

Click on Send

A screenshot of a computer

Description automatically generated with medium confidence

Every service can be implemented in the same manner.

**Using Curl**

#Go to <https://www.base64encode.org/>

Click on Encode

#Paste the key obtained from Keycloak in this format

spring-cloud-client:key obtained from keycloak

Example: spring-cloud-client:Hr9tGnoynPDpFoKAkCybri3qM3gzY3S2

Now, click on Encode.

It will generate the encoded key

A screenshot of a computer

Description automatically generated

#Copy the encoded key obtained after clicking on encode as highlighted in the image above,

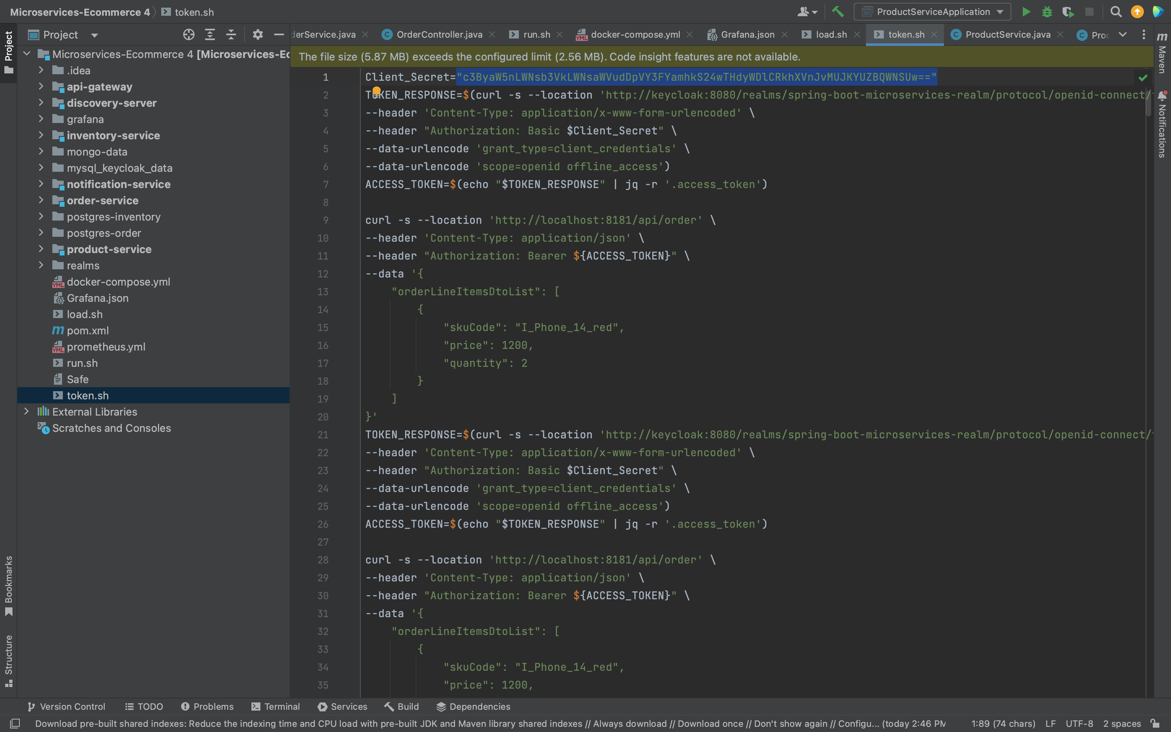
#Example:

c3ByaW5nLWNsb3VkLWNsaWVudDpIcjl0R25veW5QRHBGb0tBa0N5YnJpM3FNM2d6WTNTMg==

#Inside the ‘token.sh’ file,

#Replace the value of the Client\_Secret variable in the first line of token.sh with the encoded key.

#‘As shown in the image in the line highlighted in the image below’



On the terminal write,

./token.sh

Every service can be implemented in the this manner.

**To obtain the metrics**

#Go to your browser and type

localhost:3000

#This will open Grafana

#Login using username as **admin** and password as **password**

**A screenshot of a computer

Description automatically generated**

#Click on Add your first data source

**A screenshot of a computer

Description automatically generated**

#Select Prometheus

#Change the URL to

<http://prometheus:9090>

A screenshot of a computer

Description automatically generated

#Scroll down and click on Save and test

A screenshot of a computer

Description automatically generated

#On the left in the above image you can see a ‘+’ symbol. Click on that

A screenshot of a computer

Description automatically generated

#Click on Import

**A screenshot of a computer

Description automatically generated**

#Paste the contents of ‘Grafana.json’ shared on Github in the ‘Import via panel json’ text area visible in the image

#Click on Load

A screenshot of a computer

Description automatically generated with medium confidence

#Select Prometheus data source from Prometheus drop down and Click on the pink button ‘Import (Overwrite)’

A screenshot of a computer

Description automatically generated

And you can finally see the generated metrics

A screenshot of a computer

Description automatically generated