# Project code folder structure:

A screenshot of a cell phone

Description automatically generated

This is a maven project containing 4 modules: audit-service, gateway-service, products-service and registry-service.

# How to run local:

* Install docker engine in local, to ease to install RabbitMQ, MySQL server.
* Install MySQL: Please refer: https://dev.mysql.com/doc/mysql-installation-excerpt/8.0/en/docker-mysql-getting-started.html
* Install RabbitMQ:  
  *docker run -d --hostname my-rabbit --name rabbit-local -p 5671-5672:5671-5672 -p 15671-15672:15671-15672 -p 25672:25672 -p 4369:4369 rabbitmq:3-management*
* Build all projects in shop-online folder: mvn clean install
* For each jar file built inside target, run below command to start service, example:

*nohup java -Deureka.server.host=localhost -Deureka.server.port=1111 -jar registy-service.jar &*

* Note: These properties are required to start services:  
  **Generic properties**:

*eureka.server.host=host\_which\_registy\_service\_running (default localhost)*

*eureka.server.port=port\_ which\_registy\_service\_running (default 8761)*  
**For product-services, audit-services**:  
*spring.datasource.url= jdbc url pointing to mysql database  
spring.datasource.username= database crendential usename*

*spring.datasource.password= database crendential password  
spring.rabbitmq.host= rabbitmq server’s host*

*spring.rabbitmq.port= rabbitmq server’s port*

# CURL commands

Please provide your gateway host, gateway port to execute the commands.  
These samples below are pointing to gateway localhost:7770

* **Get all products available:**

curl -i -X GET 'http://localhost:7770/products-service/api/products/'

* **Get all categories:**

curl -i -X GET 'http://localhost:7770/products-service/api/categories/'

* **Search product by name:**

curl -i -X GET 'http://localhost:7770/products-service/api/products/search/byName?name=iphone'

* **Search product by brand:**

curl -i -X GET 'http://localhost:7770/products-service/api/products/search/byBrand?brand=apple'

* **Search product by color**:

curl -i -X GET 'http://localhost:7770/products-service/api/products/search/byColor?brand=black'

* **Search product by category:**

curl -i -X GET 'http://localhost:7770/products-service/api/products/search/byCategory?category=3'

* **To sort returned result, add parameter sort=sortBy,sortDirection (ex: sort=price,desc)**

curl -i -X GET 'http://localhost:7770/products-service/api/products/search/byName?name=iphone&sort=price,desc'

* **We can do paging result by adding parameter page and size**:

curl -i -X GET http://localhost:7770/products-service/api/products/search/byCategory?category=3&page=0&size=3