

System Requirement

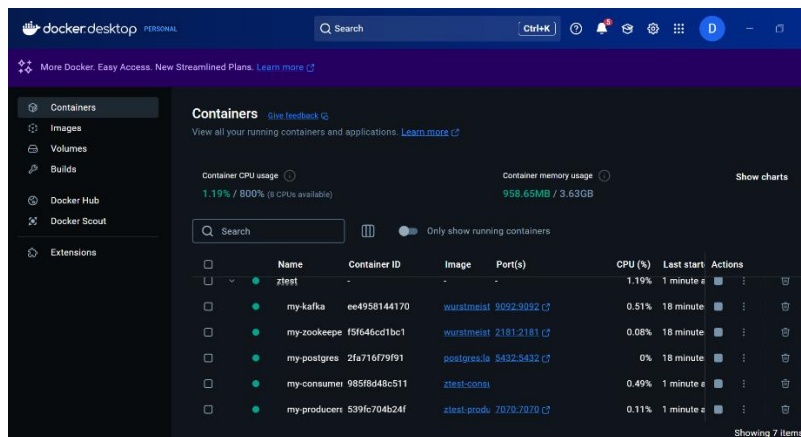
- Java with JDK 17 Version
- Maven 3.9.9 Version
- Spring Boot 3.2.5 Version
- Kafka as Message Broker
- PostgreSQL as a Database
- Docker as a Container
- Postman as a Testing Tool

Description

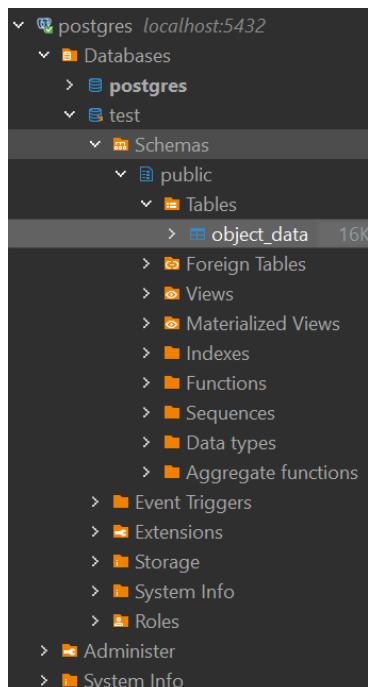
- This project have 2 services (Producers and Consumers)
- Producers Service to generate data and produce the message into kafka
- Consumers Service to retrieve from kafka and put into PostgreSQL database using JPA

How to Test

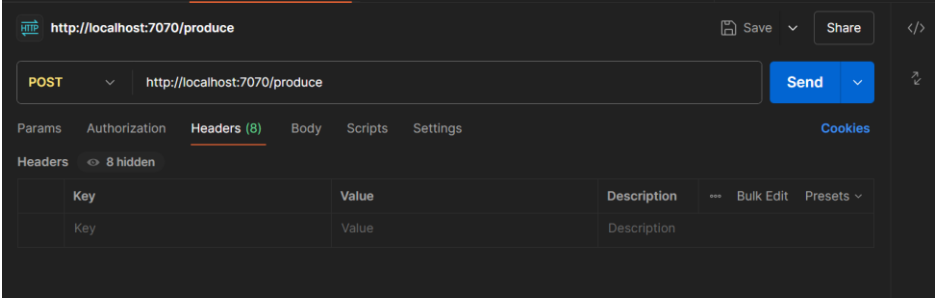
- Extract .rar file into specific path (In this case in the D:/ztest)
- After extract the file, build the application using commant ‘docker-compose up -d’ and automatic install PostgreSQL, Kafka, Zookeeper, and the Producers and Consumers Service



- The database and table automatically created while the application was running



- Run the project using this curl in the terminal or using Postman tools
curl -v <http://localhost:7070/produce>



- If success, you can get this response data and the data already filled in the database

The image shows the Postman interface displaying a successful 200 OK response. The 'Body' tab is selected, showing the JSON response:

```
{
  "code": 200,
  "status": "Success Send Data",
  "data": [
    {
      "sequence": 1,
      "messageDesc": "Product 1",
      "price": 1000
    },
    {
      "sequence": 2,
      "messageDesc": "Product 2",
      "price": 1000
    }
  ]
}
```

Below the Postman interface, there is a SQL query editor for PostgreSQL:

```
<postgres> Script-1 ×
select * from object_data;
```

Below the SQL query editor, there is a table view for 'object_data 1':

Grid	123 sequence	A-Z message_desc	123 price
1	1	Product 1	1,000
2	2	Product 2	1,000
3	3	Product 3	1,000
4	4	Product 4	1,000
5	5	Product 5	1,000
6	6	Product 6	1,000
7	7	Product 7	1,000
8	8	Product 8	1,000
9	9	Product 9	1,000
10	10	Product 10	1,000