

System Requirement

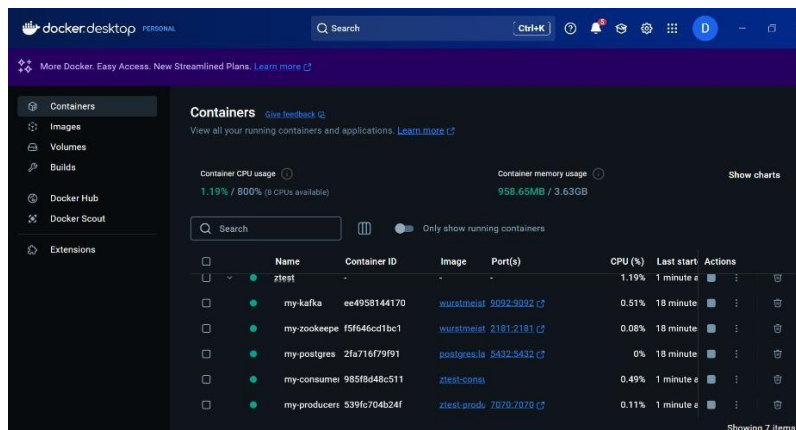
- Java with JDK 17 Version
- Maven 3.9.9 Version
- Spring Boot 3.2.5 Version
- Kafka as Message Broker
- PostgreSQL as a Database
- Docker as a Container
- Postman as a Testing Tool

Description

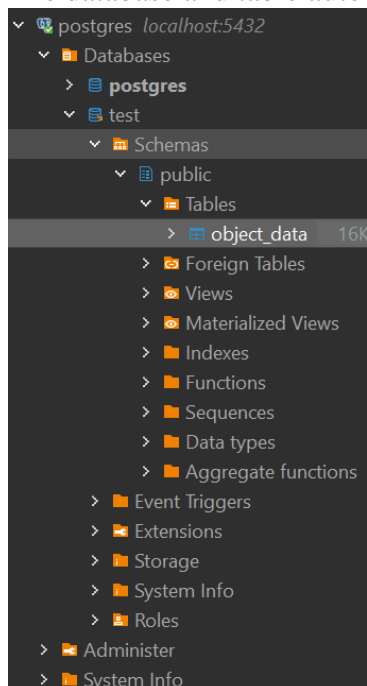
- This project have 2 services (Producers and Consumers)
- Producers Service to generate data and produce the message into kafka
- Consumers Service to retrieve from kafka and put into PostgreSQL database using JPA

How to Test

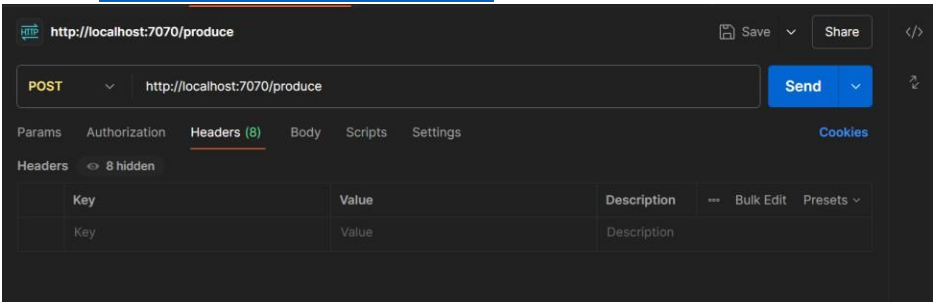
- Extract .rar file into specific path (In this case in the D:/ztest)
- After extract the file, build the application using commant 'docker-compose up -d' and automatic install PostgreSQL, Kafka, Zookeeper, and the Producers and Consumers Service



- 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 displays the Postman response for the POST request, showing a 200 OK status with a response time of 2.35 s and a body size of 756 B. The response is in JSON format, containing a success message and a list of product data.

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

Below the Postman response, a PostgreSQL query is shown in a script editor:

```
select * from object_data;
```

The query results are displayed in a table with columns: sequence, message_desc, and price. The table contains 10 rows of data, showing a sequence of products from 1 to 10, each with a price of 1,000.

Grid	sequence	message_desc	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