# Backend exercise Readme:

There are two services one is delivery service and second is automated ticketing service.

# Service 1: Delivery Service

folder name: delivery_service

Purpose: this service will publish delivery data after some specific scheduled time (only those deliveries data will be published whose full fills the conditions of ticket creation) on kafka topic **delivery.complaint.tickets**

Dummy data will load the start of service using **data.sql.** Additionally, I created a REST end point to add single delivery details. Endpoint details as follows.

URL: localhost:2021/delivery/api/v1/add

Request type: POST

| Request | Response |
|---|---|
| { "customerType": "VIP",<br>"deliveryExpectedHour":4,<br>"deliveryExpectedMins":40,<br>"timeToReachDestinationHour":4,<br>"timeToReachDestinationMins":40,<br>"meanTimeForFood":30<br>} | {"status":"success","message":"delivery added"} |

# Service 2: Automated ticketing service

folder name: ticketing

Purpose: this service will consume tickets details from kafka topic **delivery.complaint.tickets** and create ticket in database. All the tickets that are not resolved can be get by following REST endpoint.

URL: localhost:2022/callsign/api/v1/tickets/all?pageNo=0&pageSize10

Request type: GET

| Response |
|---|
| {"success": true,<br>"status": "OK",<br>"data": [ |

```
{
"ticketId": 13,
"deliverId": 7,
"customerType": "VIP",
"deliveryStatus": "0",
"deliveryRequestTime": "2022-01-12T04:59:32.706",
"expectedDeliveryCompletionTime": "2022-01-12T05:40:00",
"estimatedDeliveryCompletionTime": "2022-01-12T05:20:00",
"ticketPriority": "SEVERE",
"ticketReason": "delivery time has passed and food is not being delivered",
"ticketDetails": "",
"isTicketResolved": false
}]}
```


**All the endpoints in ticketing service are protected by JWT so you need to login first and pass bearer token in Authorization Header.**

For further ticketing service api details you can open swagger

link: **http://localhost:2022/swagger-ui.html#/**

**Test cases:**

For Unit testing I used junit with mockito and test cases of ticketing service is written and you check in test package.

**How to run Services:**

Both services have been dockerized by Dockerfile build file and all you need to do is just run **docker-compose.yml** file. You can find compose file in the zip folder.

**(note: before running docker-compose don't forget to set local time zone just need to set a property like**

**TZ:{your current time zone} e.g Asia/Dubai)**

Command For build services: **just need to run first time only**
**docker-compose build**

For run: **docker-compose up –d**

For down: **docker-compose stop**

**For spring-boot I used v2.6.2 and build type is gradle v7.2.**

**Kafka version is 2.7. For mysql is used mysql8 and docker version is 20.10.8 and docker-compose v3.8**

Total 5 images will run zookeeper, kafka, mysql, delivery_service, ticketing_service

I have try to automate complete process from build to run. There is always room for improvement but I tried to complete all the requirements that were mentioned in exercise.