

Microservice- Ecommerce

- Ankita Khurana
- 3148515
- ankita.khurana@nagarro.com

POSTMAN COLLECTION LINK : <https://documenter.getpostman.com/view/2244734/SzKPWh3y>

The postman collection describes each microservice and route function in detail.

The following is the design of the architecture made for the backend for the ecommerce web application. The services are made in **NodeJS**, the authentication is handled at API Gateway made using **Ocelot(.NET)**. The authentication is applicable on Cart, order, Payment and delivery service using **JWT token**. The service registry used: **Netflix Eureka**.

Inter microservice communication:

1. **Sync**: Cart service routes like add/delete to cart of a user requires getting product and user info synchronously, for this communication **HTTP requests** are made.
2. **Async**: Placing order, updating order status (ordered/paid/delivered), is done using **RabbitMQ**.

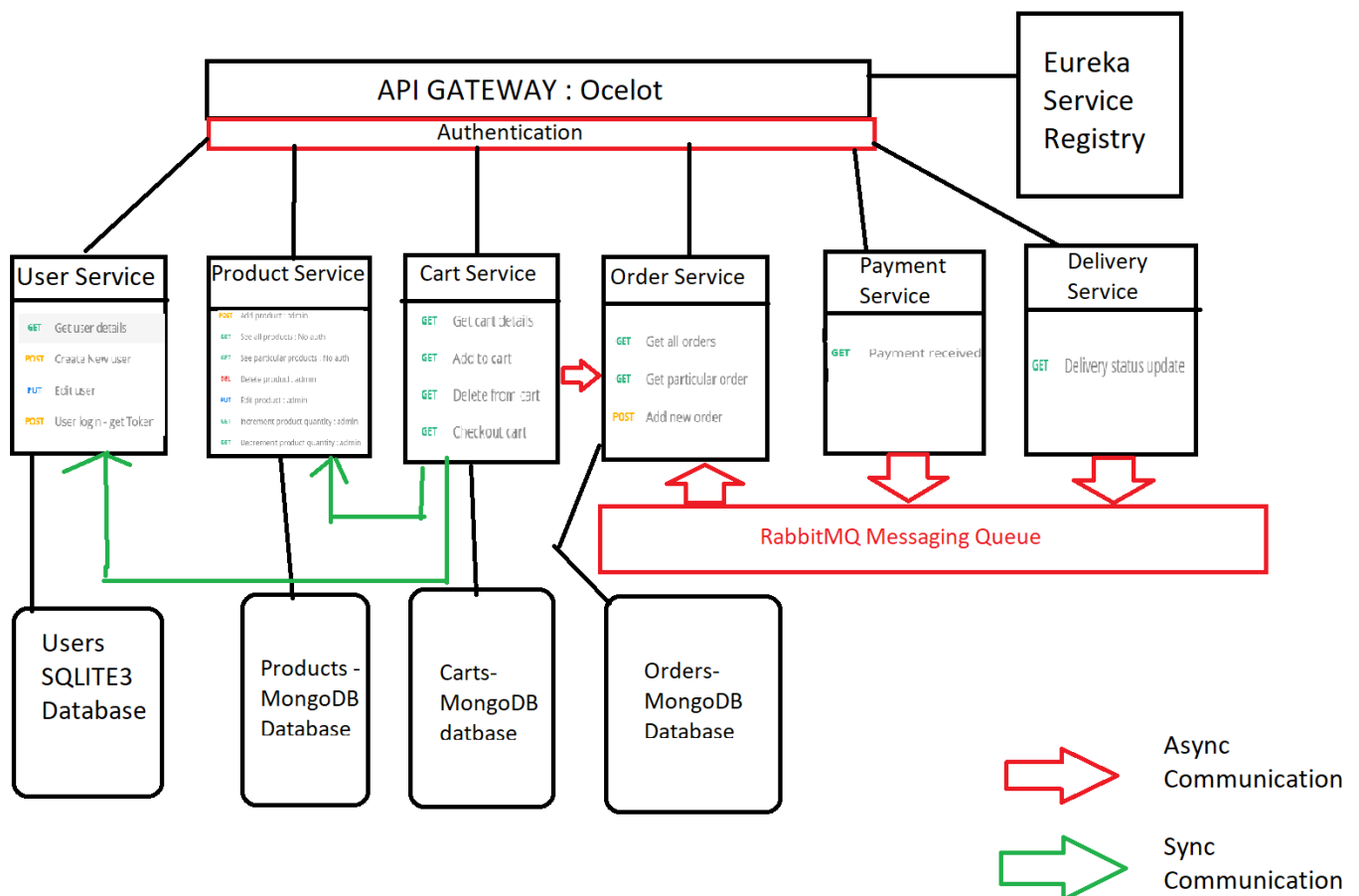
Databases used:

1. **SQLite3**: User service uses SQLite3 for maintaining user records, relational database model is used for this microservice.
2. **MongoDB**: Products, Order and Cart uses NoSQL MongoDB database for document styled storage.

Logging: **Winston** library is used for logging in microservices.

Containerization and deployment: **docker-compose.yml** file is preset in the root directory for easy deployment of API gateway, service registry eureka, databases and all microservices.

1. System Design (Architecture)



2. API GATEWAY

- Ocelot (.NET)

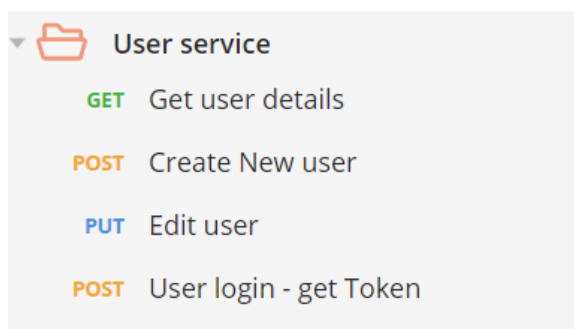
3. Service Discovery

- Netflix Eureka

2. Services: Routes

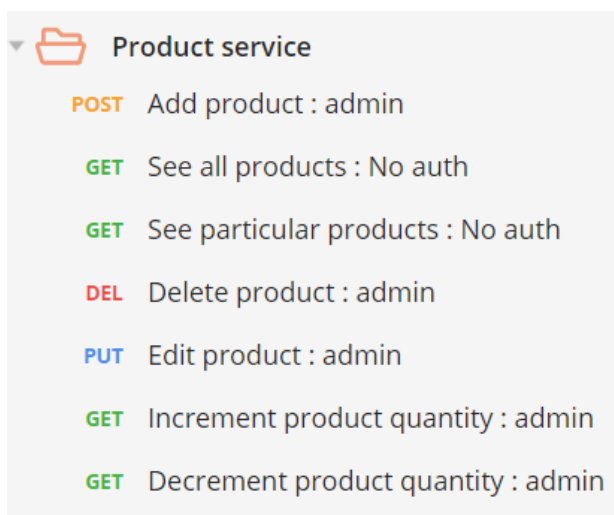
User Service

- Technology: NodeJS
- Framework used: Express.js



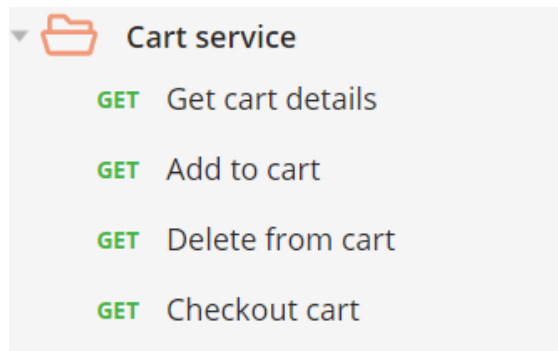
Product Service

- Technology: NodeJS
- Framework used: Express.js



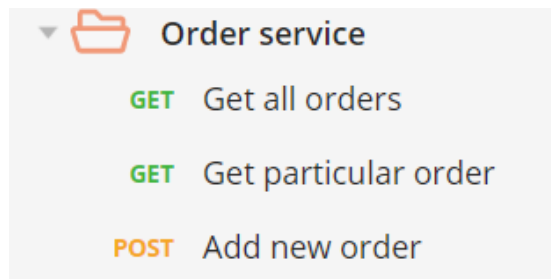
Cart Service

- Technology: NodeJS
- Framework used: Express.js



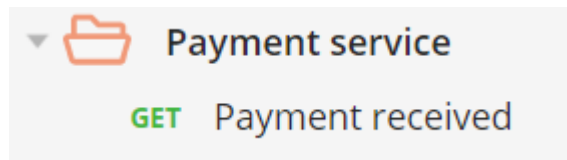
Order Service

- Technology: NodeJS
- Framework used: Express.js



Payment Service

- Technology: NodeJS
- Framework used: Express.js



Delivery Service

- Technology: NodeJS
- Framework used: Express.js

