

Retail Order Processing

Use Case

The use case of this project is the user will place the order by giving the products as inputs and we need to place the order with the order status as “placed”. And immediately with the help of messaging queue we need to update the order status as “processed”

Frameworks and Technologies Used

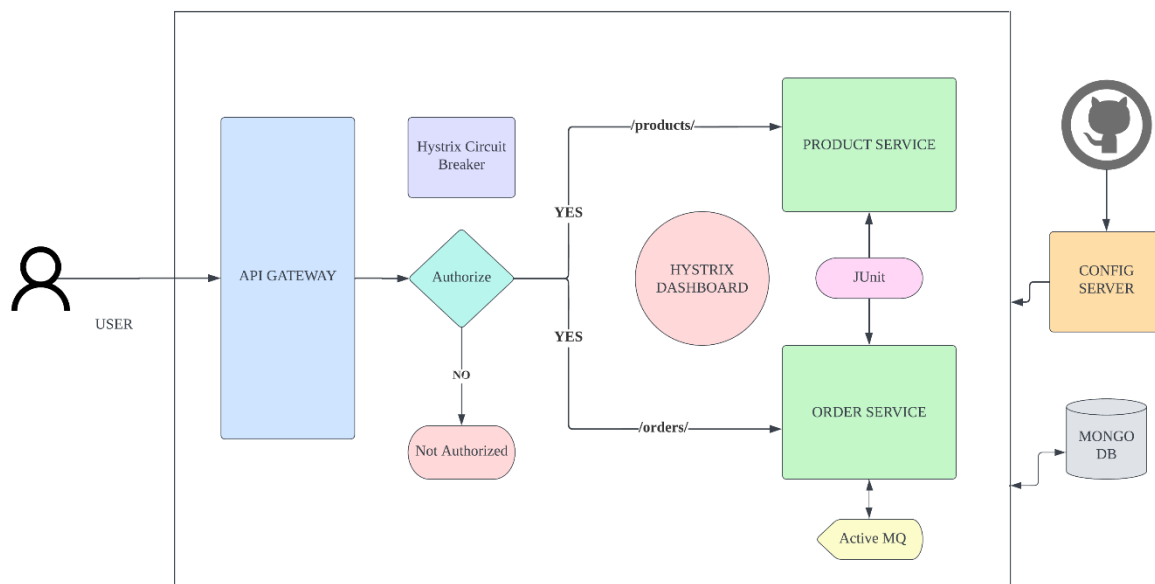
- Spring boot
- Spring cloud API Gateway
- Git
- MongoDB for database
- ActiveMQ for message driven
- Cloud Config Server for centralized configuration
- Eureka server to manage all services as eureka clients
- Hystrix for circuit breaker
- JUnit for Testing
- Auth Filter API Key for secure API
- Rest Template for service communication

Tools Used

- Spring Tool Suite
- Mongo DB Compass
- Postman

Microservices Architecture

RETAIL ORDER PROCESSING MICROSERVICES ARCHITECTURE



Microservices:

API gateway - Routes to process requests to downstream services.

Config server – Provides server-side and client-side support for externalized configuration in a distributed system

Service registry – To manage server and communicate each other micro services.

Hystrix Dashboard – To monitor the service performance

Product Service – Provides functionality manage all the product items async communication

Order Service – Provides functionality to manage all the order and async communication with order processing function.