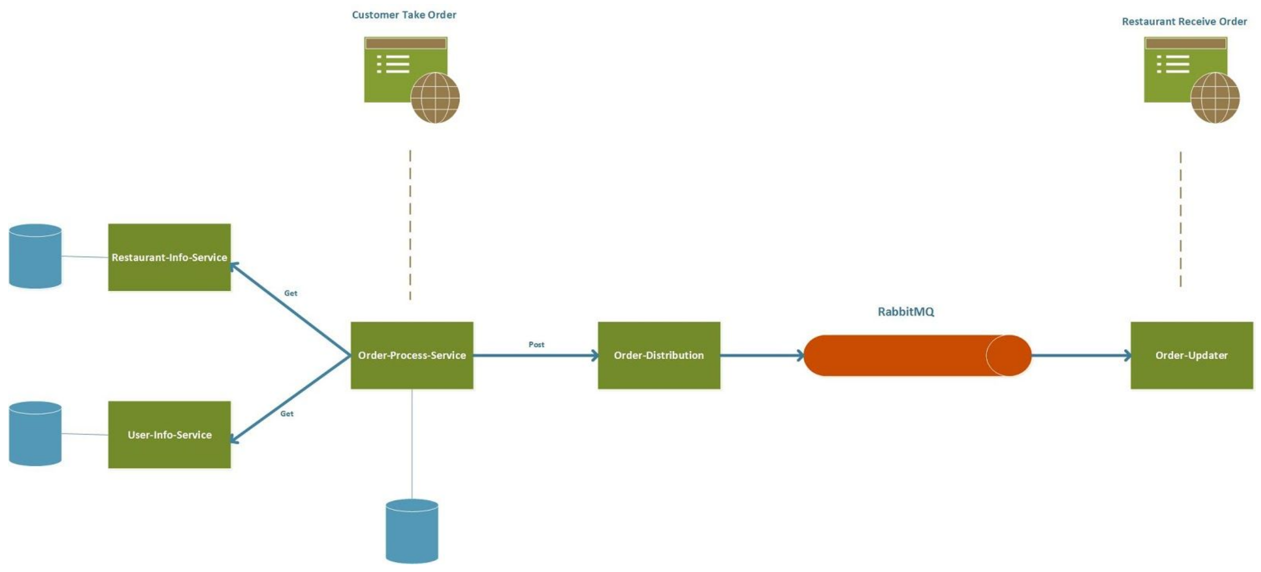
Tritoneat Design Detail

Introduction

A backend project design for online ordering platform.Customer are able to search and take order in specific restaurant. The backend side will handle order information and order process forward to the restaurant by message queue. Restaurant is able to grab success order information by websocket in the frontend web.

Project Architecture



Infrastructure

Eureka

Eureka is a REST based service that is primarily used in the cloud for locating services for the purpose of load balancing and failover of middle-tier servers. In this project, Eureka was used for microservice registry and service discovery.

- Server Information: http://localhost:8761

Hystrix

Hystrix is a latency and fault tolerance library designed to isolate points of access to remote systems, services and 3rd party libraries, stop cascading failure and enable resilience in complex distributed systems where failure is inevitable.

- Server Information: http://localhost:7979

RabbitMQ

RabbitMQ is an open source message broker software (sometimes called message-oriented middleware) that implements the Advanced Message Queuing Protocol (AMQP).

- Server Information: http://localhost:5672
- Server management web portal: http://localhost:15672

User-Info-Serivce

- Provide API and search function for all users information.
- Server Information: http://localhost:8001
- Rest API
  - POST:/userinfo/post
  - DELETE:/userinfo/delete
  - GET:
    - FindAll:/userinfo/get
    - FindByUserName:/userinfo/get/name/{userName}

Restauarnt-Info-Serivce

- Provide API and search function for all restaurant information.
- Server Information: http://localhost:8002
- Rest API
  - POST: /restaurantinfo/post

        - DELETE:/restaurantinfo/delete
        - GET:
            - FindAllRestaurantInfo: /restaurantinfo/get
            - FindByRestaurantName: /restaurantinfo/get/name/{restaurantName}
            - FindByRestaurantType:/restaurantinfo/get/type/{restaurantType}
            - FindByRestaurantId::/restaurantinfo/get/id/{restaurantId}

Order-Process-Service

When customer take order, order process service will get more detail information from both user-info and restaurant-info. After validate payment and restaurant information, it will create order and post to order-distribution service.

- Server Information: http://localhost:8003
- Customer search and take order page: http://localhost:8003
- Rest API
    - POST: /orderinfo/post
    - DELETE:/orderinfo/delete
    - GET:
        - FindAll:/orderinfo/get

Order-Distribution

Order distribution service will get the order from Order-process-Service and put the order into the right message queue. In this project, we only have one message queue running in RabbitMQ.

- Server Information: http://localhost:8004

Order-Updater

Order Updater will receive order information from message queue and websocket will forward order to front-end. Restaurant is able to grab success order information by websocket in the frontend web.

- Server Information: http://localhost:8005
- Restaurant side web port: http://localhost:8005