Создайте систему обмена сообщениями, используя RabbitMQ в связке со Spring Boot, где сообщения будут отправляться в зависимости от региона. Проект состоит из двух микросервисов:

1. **Order Service** — Сервис для создания заказов.
2. **Notification Service** — Сервис, который получает уведомления о новых заказах и изменения их статуса.

 **Order Service:**

* Создайте проект Spring Boot с зависимостью spring-boot-starter-amqp.
* Настройте application.properties для подключения к RabbitMQ.
* Определите OrderDTO класс, который будет содержать данные заказа, включая поля для ресторана, курьера и списка блюд.
* Настройте topic exchange с именем order-topic-exchange для отправки заказов в разные очереди на основе ключа маршрутизации (например, order.almaty, order.astana и т.д.).
* Создайте OrderPublisher, который отправляет объект OrderDTO в обменник order-topic-exchange с ключом маршрутизации, соответствующим региону.

 **Notification Service:**

* Создайте проект Spring Boot с зависимостью spring-boot-starter-amqp.
* Настройте application.properties аналогично Order Service, но добавьте настройку очередей для разных регионов.
* Создайте OrderNotificationListener, который будет прослушивать сообщения с order-topic-exchange по ключу маршрутизации order.# и выводить в лог полученные заказы.

**Инструкции по выполнению**

1. **Docker**: Запустите RabbitMQ с помощью команды docker-compose up.
2. **Запуск Order Service**: Запустите сервис и отправьте заказ через POST запрос в OrderController с указанием региона. Например:

json

Копировать код

{

"restaurant": "Example Restaurant",

"courier": "Courier Name",

"foods": ["Pizza", "Burger"],

"status": "New"

}

1. **Запуск Notification Service**: Запустите сервис и проверьте, что он получает сообщения в зависимости от региона. Для тестирования отправьте несколько заказов в разные регионы и убедитесь, что они попадают в соответствующие очереди.