

## Yêu cầu:

Giả sử bạn có một hệ thống đặt hàng từ xa. Client thông qua ứng dụng (desktop/web) gửi thông tin đặt hàng sau khi chọn lựa. Thông tin này được chuyển về dạng json và được mã hóa (có thể dùng public/private keys hoặc mã hóa theo mật khẩu hay 1 loại đơn giản nào đó)

Để tăng hiệu suất xử lý, hệ thống sử dụng messaging service cho việc lắng nghe việc đặt hàng. Khi nhận được đơn hàng từ client, hệ thống sẽ kiểm tra số lượng trong kho có đủ không (có thể mở rộng cho trường hợp kiểm tra dịch vụ giao nhận theo địa chỉ, kiểm tra năng lực thanh toán,...) sau đó quyết định đơn hàng có được xác nhận hay không. Cho dù xác nhận hay hủy bỏ, hệ thống cũng sẽ gửi một email đến khách hàng để thông báo tình hình.

## Hướng dẫn:

1. Tạo csdl cho việc lưu trữ hàng đơn giản. Có thể bao quát các yêu cầu trong 1 bảng *product* (với lý do là bài tập đơn giản. Thực tế thì không được làm như thế mà cần nhiều vấn đề liên quan.)  
Một bảng *product\_order* để lưu thông tin đặt hàng (mã kh, ....)
2. Tải ActiveMQ, cấu hình và chạy broker.
3. Sử dụng spring data để viết các lớp theo *kiến trúc layer* để truy xuất csdl và thực hiện các yêu cầu.
4. Tạo 1 dự án sử dụng spring-boot với dependency đến **spring-boot-starter-activemq**.

Cấu hình cho activemq – application.properties:

```
spring.activemq.broker-url=tcp://localhost:61616
spring.activemq.user=admin
spring.activemq.password=admin
```

Viết code cho việc lắng nghe trên một queue (name *order\_products*)

```
import com.google.gson.Gson;
import org.springframework.jms.annotation.JmsListener;
import org.springframework.messaging.handler.annotation.SendTo;
import org.springframework.stereotype.Component;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.TextMessage;
import java.util.Map;

@Component
public class ProductOrderListener {

    @JmsListener(destination = "order_products")
    public void receiveMessage(final Message jsonMessage) throws JMSException {
        String messageData = null;
        String response = null;
        if(jsonMessage instanceof TextMessage) {
            //1. read message data
            //2. ==> decode
            //3. check for quantity
            //4. make order or reject
            //5. send email
        }
    }
}
```

Tham khảo về ActiveMQ với Springboot:

<https://docs.spring.io/spring-boot/docs/3.3.x/reference/html/messaging.html#messaging>

New Project

Empty Project

Generators

Maven Archetype

Jakarta EE

Spring Initializr

JavaFX

Quarkus

Micronaut

Ktor

Compose for Desktop

HTML

React

Express

Angular CLI

Vue.js

Vite

Server URL: [start.spring.io](https://start.spring.io)

Name:

Location:

Project will be created in: T:\VoVanHai\source-code\layer-demo

☒ Create Git repository

Language:

Type:

Group:

Artifact:

Package name:

JDK:

Java:

Packaging:

?

Next

Cancel

#### Added dependencies:

- × Spring Boot DevTools
- × Lombok
- × Spring Web
- × Thymeleaf
- × Spring Data JPA
- × MariaDB Driver
- × Spring for Apache ActiveMQ 5

