

Một công ty cần một hệ thống xử lý đơn hàng như mô tả sau

Khi một khách hàng có yêu cầu đặt hàng, họ chọn các mặt hàng và số lượng hàng cần mua. Ngoài ra, họ còn cần cung cấp thông tin về địa chỉ nhận hàng, số tài khoản cùng ngân hàng để xác nhận năng lực thanh toán. Sau đó gửi yêu cầu về công ty. Những thông tin này được mã hóa trong một file json có cấu trúc như sau

```
{
  "order_info": {
    "orders": [
      {
        "orderId": 1,
        "custId": 1,
        "productId": 1,
        "quantity": 100,
        "description": "sample order 1"
      },
      {
        "orderId": 2,
        "custId": 1,
        "productId": 2,
        "quantity": 10,
        "description": "sample order 2"
      }
    ],
    "delivery": {
      "nodeId": 1,
      "note": "sample note 1",
      "deliveryAdd": "12 NVB F4 GV, Ho Chi Minh, 76001, Viet Nam",
      "isDelivery": false
    },
    "payments": {
      "paymentId": 1,
      "custId": 1,
      "cardNumber": "7485-2222-3456-2435",
      "cvv": "111"
    }
  }
}
```

Hãy xây dựng một hệ thống dùng pipeline arch xử lý vấn đề này. Hệ thống bao gồm các tasks:

- Kiểm tra hàng đủ cung ứng
- Kiểm tra dịch vụ giao nhận và cước phí
- Kiểm tra năng lực thanh toán
- Lập đơn đặt hàng
- Gửi email thông báo.

Để có thể xử lý liên tục dữ liệu, hệ thống cần một message queue nhận các đơn hàng này rồi phân phối cho các module xử lý.

Ngoài ra, về mặt giả lập, hãy viết một service gửi đơn hàng liên tục (thời gian random) về một queue với cấu trúc sinh kết quả như mẫu tài liệu json. Các filter sẽ nhận tài liệu rồi xử lý theo các tác vụ tương ứng.

Các vấn đề/ giả thuyết gợi ý khác:

- Tập các mặt hàng là xác định (50 mặt hàng)

- Tập khu vực giao nhận là xác định (100). Viết dịch vụ xử lý giao nhận mô phỏng trả về giá tiền và thời gian nhận hàng.
- Viết dịch vụ mô phỏng một ngân hàng đơn giản với dịch vụ cho phép kiểm tra năng lực thanh toán và thanh toán.
- Viết trang web đơn giản cho khách hàng chọn hàng.