Tài liệu Spring boot + RabbitMQ

Tài liệu bao gồm:

1. Giới thiệu RabbitMQ
    1.1. Khái niệm và lợi ích
    1.2. Các thuật ngữ, thành phần chình trong RabbitMQ
    1.3. Minh họa luồng tin nhắn trong RabbitMQ (Message flow)
2. Spring boot + RabbitMQ (example)
3. Chú ý khi áp dụng với Spring boot
4. Nguồn tham khảo

---

1. Giới thiệu RabbitMQ

1.1. Khái niệm và lợi ích

- RabbitMQ hay còn được gọi là message-queueing, message broker, queue manager. Nói 1 cách dễ hiểu RabbitMQ là 1 phần mềm giúp các ứng dụng kết nối để truyền tin nhắn (người đưa thư)

- RabbitMQ rất có ích cho các dự án được triển khai theo kiến trúc Microservice vì sự đa dạng về ngôn ngữ lập trình của kiểu kiến trúc này. RabbitMQ giúp ta không cần phải lo về việc giao tiếp, trao đổi data giữa các module (có thể được bằng C#, java, …).

- Tuy nhiên việc áp dụng RabbitMQ khá phức tạp, vì vậy nên cân nhắc khi áp dụng cho những dự án không phải đa ngôn ngữ lập trình.

1.2. Các thuật ngữ, thành phần chính trong RabbitMQ

- Producing: hành động gửi message, 1 chương trình gửi tin được gọi là Producer

- Consuming: hành động nhận message, 1 chương trình nhận tin được gọi là Consumer

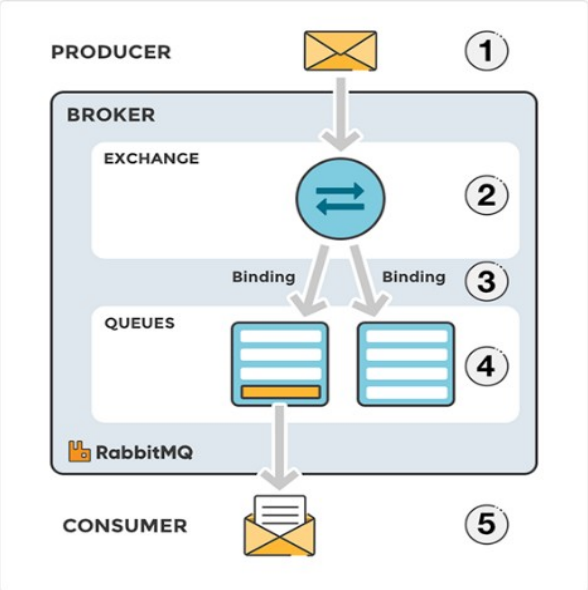
- Queue: nơi chứa message (giống hòm thư). Producer sẽ gửi mesage lên đây (thật ra thường gửi cho Exchange trước) và tương ứng là Consumer sẽ nhận message ở đây

(note: 3 thằng trên không nhất thiết phải ở cùng 1 host)

- Connection: giao thức TCP giữa RabbitMQ và ứng dụng
- Chanel: kết nối ảo trong Connection

- Binding: liên kết giữa Exchange và Queue
- Routing key: khóa giúp Exchange xác định Queue
- Exchange: có nhiệm vụ định tuyến message đến queue tương ứng. Có 4 loại Exchange:
    + DirectExchange
    + FanoutExchange
    + TopicExchange
    + HeaderExchange

1.3. Minh họa luồng tin nhắn trong RabbitMQ (Message flow)



2. Spring boot + RabbitMQ (example)

- Ví dụ gồm 2 module: 1 module đại diện cho producer và 1 đại diện cho consumer

- Producer: cần cấu hình để binding tới Exchange, RabbitTemplate để gửi message
- Consumer: cần chỉ ra queue sẽ nghe bằng chú thích @RabbitListener

- Trong ví dụ này producer sẽ gửi message là 1 object (Company) lên queue và consumer sẽ được cấu hình để lắng nghe trên queue này.

- Để gửi được object (dạng JSON) thông qua RabbitMQ cần phải convert message bằng phương thức setMessageConverter(MessageConverter). Đối với Producer thì set cho RabbitTemplate còn Consumer thì set cho SimpleRabbitListenerContainerFactory.

- link source code: *https://github.com/tuantvgithub/osp-task/tree/29-07-2021*

3. Chú ý khi áp dụng với Spring boot

- Lỗi liên quan đến quan hệ giữa các Entity (OneToMany, …)
  → giải pháp: có thể dùng @JsonIdentityInfo cho các Entity

- Lỗi tạo các Bean liên quan đến cấu hình RabbitMQ (ví dụ RabbitTemplate, AmqpTemplate, …)
  → giải pháp: lỗi này là do chúng được tự động tạo nên nếu muốn tạo lại (ghi đè) thì thêm thuộc tính ghi đè vào tệp application.properties

- Có thể cần implements Serializable phục vụ cho việc convert message và có thể cần @JsonProperty cho các thuộc tính

4. Nguồn tham khảo

- RabbitMQ:
https://www.cloudamqp.com/blog/part1-rabbitmq-for-beginners-what-is-rabbitmq.html

- Spring boot + RabbitMQ (code example):
https://www.javainuse.com/spring/spring-boot-rabbitmq-hello-world

https://grokonez.com/spring-framework/spring-amqp/rabbitmq-sendreceive-java-object-messages-spring-rabbitmq-springboot