



CSC12108 – ỨNG DỤNG PHÂN TÁN

HƯỚNG DẪN THỰC HÀNH

Giao tiếp bất đồng bộ

I. Thông tin chung

Mã số:	HD01
Thời lượng dự kiến:	3 tiếng
Deadline nộp bài:	-
Hình thức:	-
Hình thức nộp bài:	-
GV phụ trách:	Phạm Minh Tú
Thông tin liên lạc với GV:	pmtu@fit.hcmus.edu.vn

II. Chuẩn đầu ra cần đạt

Bài hướng dẫn này nhằm mục tiêu đạt giúp sinh viên được các mục tiêu sau:

- Sử dụng mã nguồn mở message broker **RabbitMQ** để cài đặt cơ chế giao tiếp bất đồng bộ
- Xây dựng ứng dụng phía publisher để gửi một message
- Xây dựng ứng dụng phía subscriber để nhận một message

III. Mô tả

a. Môi trường phát triển

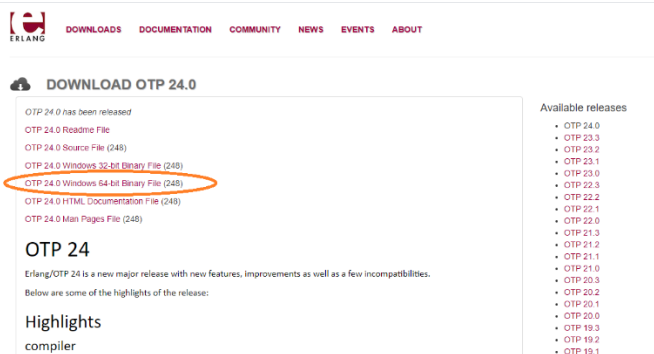
Trong bài này, các công cụ và môi trường dưới đây sẽ được dùng để hướng dẫn sinh viên thực hành môn học.

- **Eclipse IDE for Enterprise Java Developers.**
 - Version: 2019-03 (4.11.0)
 - Build id: 20190314-1200
- **Java JDK 1.8**

b. Cài đặt RabbitMQ

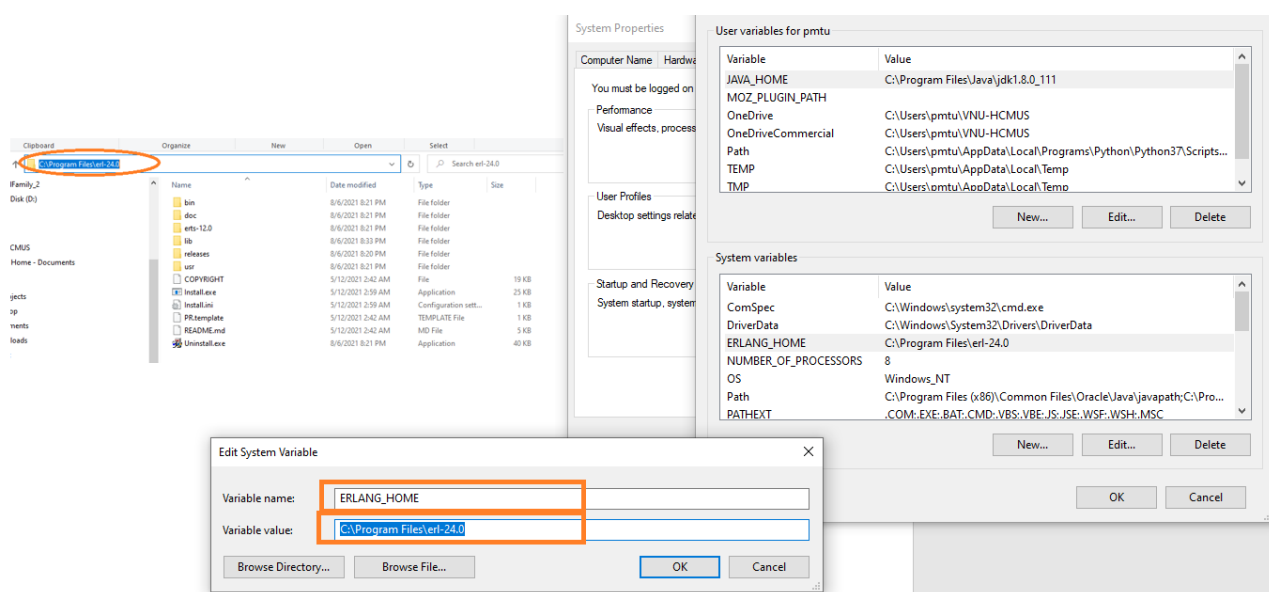
Bước 1: Tải **ERLang** và cài đặt theo hướng dẫn

<https://www.erlang.org/downloads>



Sau khi cài đặt, tiến hành đặt biến môi trường như sau:

Xác định đường dẫn thư mục cài đặt: **erlang**



Bước 2: Cài đặt RabbitMQ

<https://www.rabbitmq.com/install-windows-manual.html>

Install RabbitMQ Server

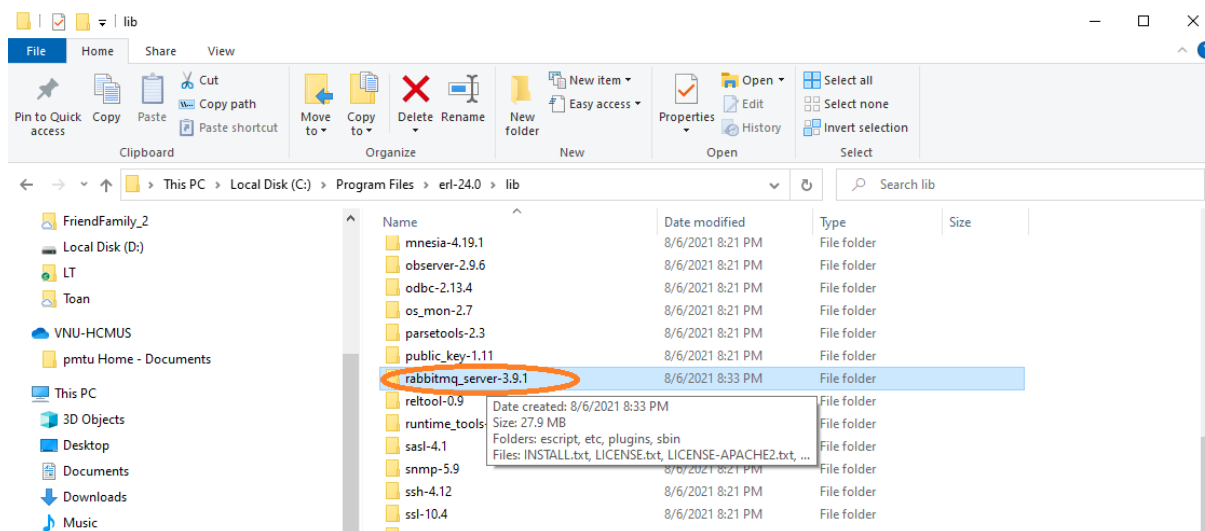
After making sure a supported Erlang version is installed, download **rabbitmq-server-windows-3.9.1.zip**.

Direct Downloads

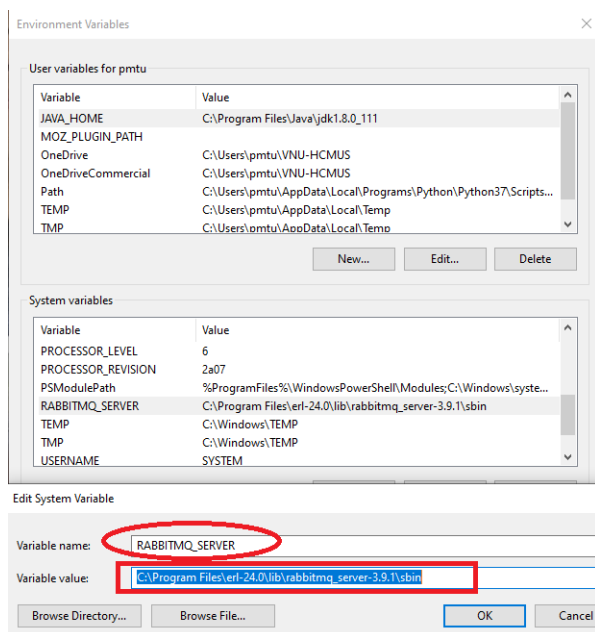
Description	Download	Signature
Installer for Windows systems (from GitHub)	rabbitmq-server-windows-3.9.1.zip	Signature

Sau khi tải về, giải nén và đặt vào thư mục **lib** của đường dẫn cài đặt Erlang

C:\Program Files\erl-24.0\lib



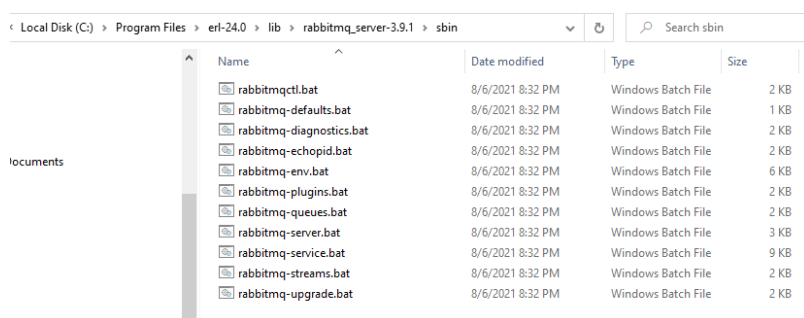
Tiến hành cài đặt biến môi trường cho RabbitMQ



c. Khởi động message broker RabbitMQ

Tìm đến vị trí đường dẫn rabbitmq và mở CMD

C:\Program Files\erl-24.0\lib\rabbitmq_server-3.9.1\sbin



Gõ lệnh: *rabbitmq-server.bat*



Mở một CMD thứ 2 và gõ lệnh

```
rabbitmq-plugins enable rabbitmq_management --online
```

Lệnh trên sẽ mở plugin quản lý rabbitmq bằng ứng dụng web admin

Tiếp theo mở trình duyệt và gõ địa chỉ:

<http://localhost:15672/>

Logic với tài khoản: guest/guest

The image shows the RabbitMQ login interface. It features the RabbitMQ logo at the top. Below it, there are two input fields: 'Username' with the value 'guest' and 'Password' with masked characters '.....'. To the right of each field is a red asterisk. Below the password field is an orange 'Login' button.

Tạo exchange

The image shows the RabbitMQ web interface for managing exchanges. The 'Exchanges' tab is selected and highlighted with a red circle. Below the navigation bar, there is a table of existing exchanges. At the bottom, the 'Add a new exchange' form is shown, with a red box highlighting the input fields and the 'Add exchange' button. A red arrow points to the 'Type' dropdown menu.

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
amq.direct	direct	D			
amq.fanout	fanout	D			
amq.headers	headers	D			
amq.match	headers	D			
amq.rabbitmq.trace	topic	D I			
amq.topic	topic	D			

Add a new exchange

Name: *

Type: ←

Durability:

Auto delete:

Internal:

Arguments: =

Exchange là tác nhân định hướng các message đến các hàng đợi tương ứng.



Ví dụ ở trên tạo exchange có tên : my_exchange

Tiếp theo sẽ tạo các hàng đợi tương ứng và liên kết chúng với exchange vừa tạo. Số lượng hàng đợi không giới hạn.

RabbitMQ™ RabbitMQ 3.9.1 Erlang 24.0

Overview Connections Channels Exchanges **Queues** Admin

Queues

▼ All queues (0)

Pagination

Page 1 of 0 - Filter: ☐ Regex ?

... no queues ...

▼ Add a new queue

Type: Classic ▼

Name: queue_1 *

Durability: Durable ▼

Auto delete: ? No ▼

Arguments: = String ▼

Add Message TTL ? Auto expire ? Overflow behaviour ? Single active consumer ?

Dead letter exchange ? Dead letter routing key ?

Max length ? Max length bytes ?

Maximum priority ? Lazy mode ? Master locator ?

Add queue

Ví dụ tạo 3 hàng đợi như sau:

RabbitMQ™ RabbitMQ 3.9.1 Erlang 24.0

Overview Connections Channels Exchanges **Queues** Admin

Queues

▼ All queues (3)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Overview				Messages			Message rates				+/-
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver	/ get	ack	
queue_1	classic	D Args	idle	0	0	0					
queue_2	classic	D Args	idle	0	0	0					
queue_3	classic	D Args	idle	0	0	0					

Chọn từng hàng đợi để liên kết đến exchange vừa tạo ở trên

Overview				Messages			Message rates				+/-
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver	/ get	ack	
<u>queue_1</u>	classic	D Args	idle	0	0	0					
queue_2	classic	D Args	idle	0	0	0					
queue_3	classic	D Args	idle	0	0	0					



▼ Bindings

From	Routing key	Arguments
(Default exchange binding)		

⇓

This queue

Add binding to this queue

From exchange: my_exchange *

Routing key:

Arguments: = String ▼

Bind

Kết quả:

RabbitMQ™ RabbitMQ 3.9.1 Erlang 24.0

Overview Connections Channels **Exchanges** Queues Admin

Exchange: my_exchange

▼ Overview

Message rates last minute ?

Currently idle

Details

Type	fanout
Features	durable: true
Policy	

▼ Bindings

This exchange

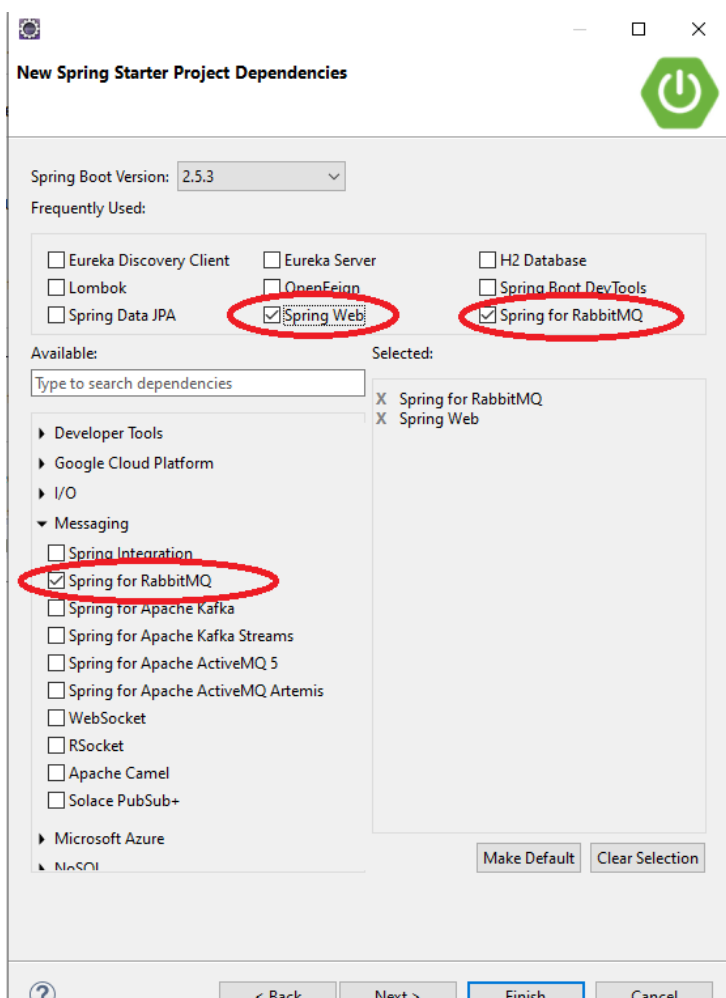
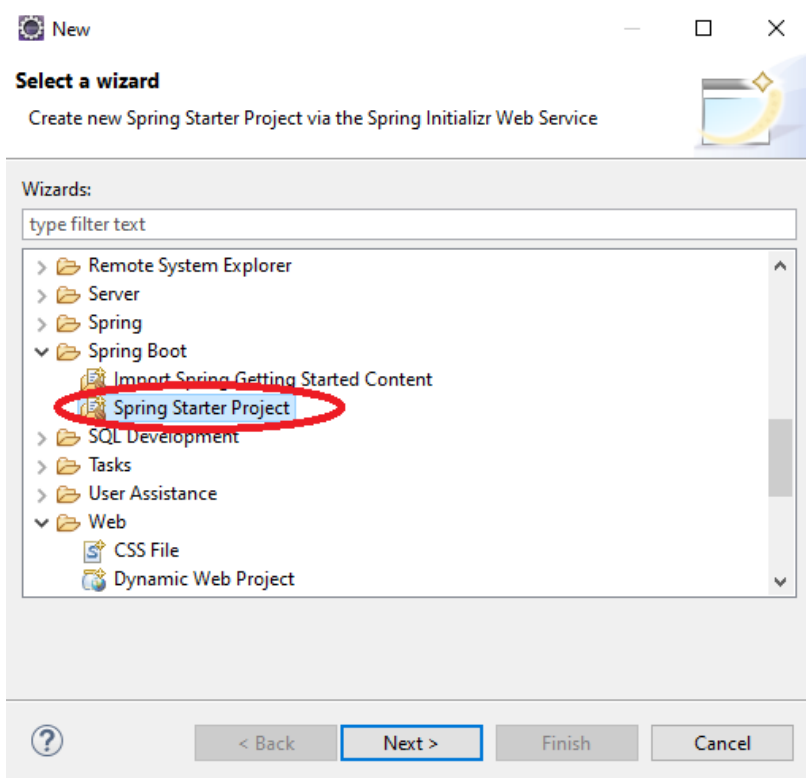
⇓

To	Routing key	Arguments	
queue_1			Unbind
queue_2			Unbind
queue_3			Unbind

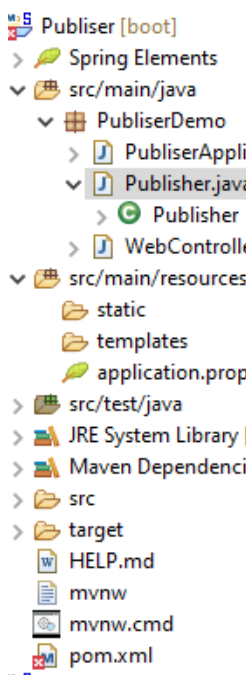
Bước 3: Tạo Project đóng vai trò người gửi message (Producer hoặc Publisher)

Mục tiêu minh họa mô hình **Publisher/Subscriber**

Tạo Project và chọn các thư viện rabbitmp như hình bên dưới:



Cấu trúc project như hình bên dưới



Lớp publisher được dùng để gửi message đến MessageBroker (rabbitMQ)

API để người dùng cung cấp nội dung message

Cấu hình các thông số cho rabbitMQ để dễ dàng sử dụng

Nội dung các classes và các file cấu hình như sau:

```

@Component
public class Publisher {

    @Autowired
    private AmqpTemplate amqpTemplate;

    @Value("${jsa.rabbitmq.exchange}")
    private String exchange;

    public void produceMsg(String msg){
        amqpTemplate.convertAndSend(exchange, "", msg);
        System.out.println("Send msg = " + msg);
    }

}
  
```

```

@RestController
public class WebController {

    @Autowired
    Publisher publisher;

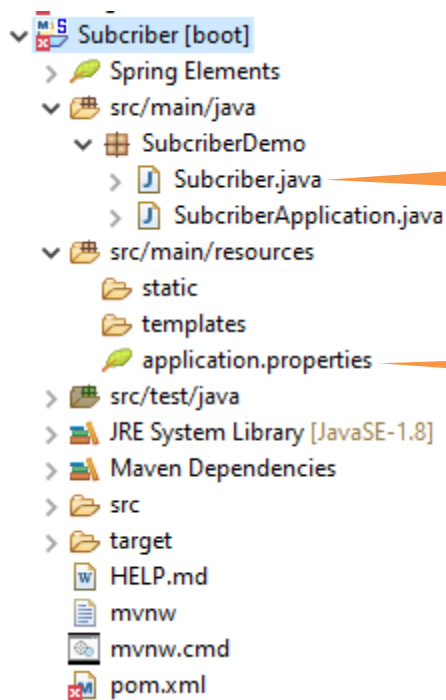
    @RequestMapping("/send")
    public String sendMsg(@RequestParam("msg")String msg){
        publisher.produceMsg(msg);
        return "Done";
    }

}
  
```



```
spring.rabbitmq.host=localhost
spring.rabbitmq.port=5672
spring.rabbitmq.username=guest
spring.rabbitmq.password=guest
jsa.rabbitmq.exchange=my_exchange
```

Tương tự, hãy tạo một project khác đóng vai trò consumer/subscriber



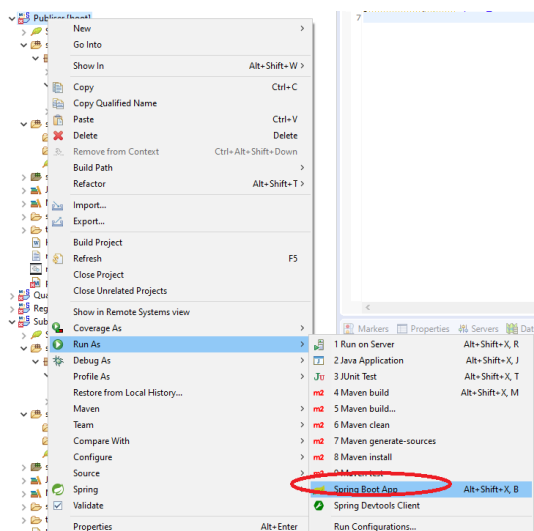
Lớp này dùng để nhận message từ message broker. Kết quả nhận được tức thì nếu message tồn tại trên hàng đợi của message broker

Cấu hình thông tin message broker, tương tự như publisher, chỉ khác cần chỉ rõ hàng đợi nào cần nhận message

```
@Component
public class Subscriber {
    @RabbitListener(queues="${jsa.rabbitmq.queue}")
    public void recievedMessage(String msg) {
        System.out.println("Recieved Message: " + msg);
    }
}
```

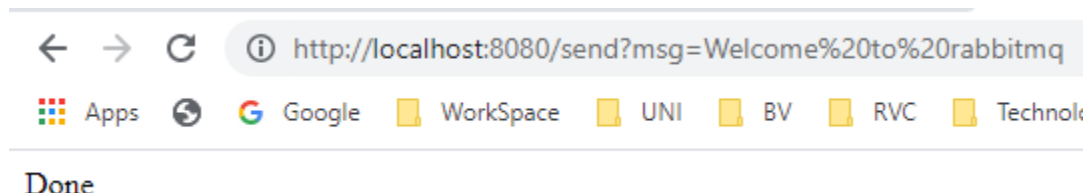
```
spring.rabbitmq.host=localhost
spring.rabbitmq.port=5672
server.port = 8082
spring.rabbitmq.username=guest
spring.rabbitmq.password=guest
jsa.rabbitmq.queue=queue_1
```

Triển khai Publisher



Gửi message bằng cách gọi API theo url như sau:

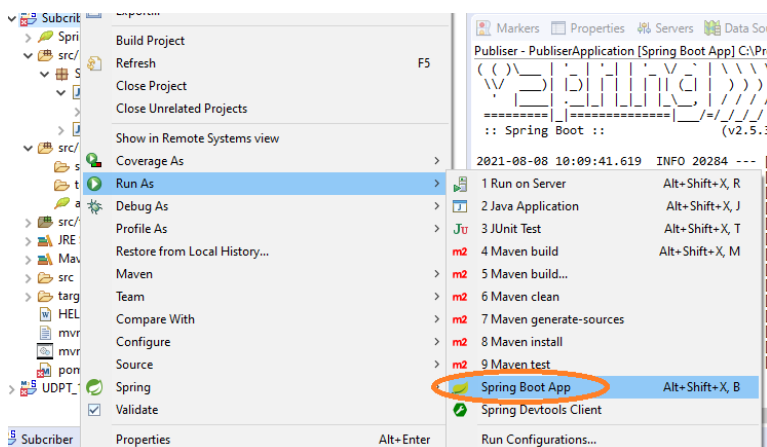
<http://localhost:8080/send?msg=Welcome> to rabbitmq



Kiểm tra message broker, chúng ta thấy các hàng đợi đã lưu lại message (Welcome to rabbitmq)

Overview				Messages			Message rates			+/-
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
queue_1	classic	D Args	idle	1	0	1	0.00/s			
queue_2	classic	D Args	idle	1	0	1	0.00/s			
queue_3	classic	D Args	idle	1	0	1	0.00/s			

Tiếp theo, triển khai Subscriber, các message trên message broker sẽ gửi về subscriber theo hàng đợi liên kết và xóa message trên hàng đợi đó.



```

Subscriber - SubscriberApplication [Spring Boot App] - C:\Program Files\Java\jdk-11.0.11\bin\java.exe (Aug 9, 2021, 10:13:27 AM)

:: Spring Boot ::
(v2.5.3)

2021-08-08 10:13:38.913 INFO 9228 --- [main] SubscriberDemo.SubscriberApplication : Starting SubscriberApplication using Java 11.0.11 on DESKTOP-7BL33QP with PID 9228 (D:\workspace_ec
2021-08-08 10:13:39.754 INFO 9228 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2021-08-08 10:13:39.762 INFO 9228 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2021-08-08 10:13:39.826 INFO 9228 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.50]
2021-08-08 10:13:39.826 INFO 9228 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2021-08-08 10:13:39.826 INFO 9228 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 855 ms
2021-08-08 10:13:40.630 INFO 9228 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8081 (http) with context path ''
2021-08-08 10:13:40.632 INFO 9228 --- [main] o.s.a.r.c.CachingConnectionFactory : Attempting to connect to: [localhost:5672]
2021-08-08 10:13:40.664 INFO 9228 --- [main] o.s.a.r.c.CachingConnectionFactory : Created new connection: rabbitConnectionFactory#40368a46:0/SimpleConnection@65eabaab [delegate=amqp:
2021-08-08 10:13:40.711 INFO 9228 --- [main] SubscriberDemo.SubscriberApplication : Started SubscriberApplication in 2.148 seconds (JVM running for 2.838)

Received Message: Welcome to rabbitmq
    
```

Message đã gửi đến Subscriber

Kiểm tra lại message broker, ta thấy message đã bị xóa khi gửi thành công đến subscriber

Overview				Messages			Message rates			+/-
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
queue_1	classic	D Args	idle	0	0	0	.00/s	0.00/s	0.00/s	
queue_2	classic	D Args	idle	1	0	1	0.00/s			
queue_3	classic	D Args	idle	1	0	1	0.00/s			

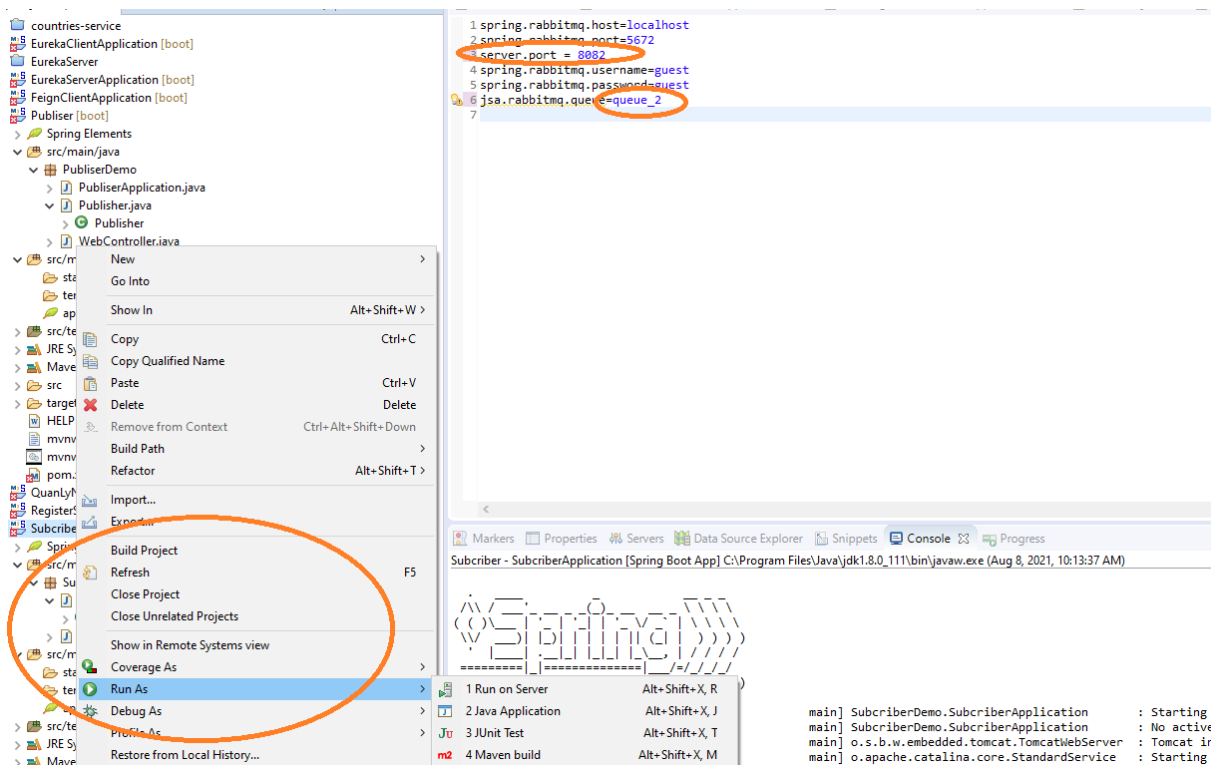
Để rõ hơn, hãy thay đổi giá trị hàng đợi trên subscriber để nhận lại các message từ message broker

```

1 spring.rabbitmq.host=localhost
2 spring.rabbitmq.port=5672
3 server.port = 8081
4 spring.rabbitmq.username=guest
5 spring.rabbitmq.password=guest
6 jsa.rabbitmq.queue=queue_1
7
    
```

Thay thế các hàng đợi khác trên message broker. Ví dụ: queue_2, queue_3

Sau đó triển khai lại Subscriber, các message từ hàng đợi queue_2 và queue_3 cũng sẽ gửi đến Subscriber



```

1 spring.rabbitmq.host=localhost
2 spring.rabbitmq.port=5672
3 server.port = 8082
4 spring.rabbitmq.username=guest
5 spring.rabbitmq.password=guest
6 jsa.rabbitmq.queue=queue_2
7
    
```

Run As

```

main] SubscriberDemo.SubscriberApplication : Starting
main] SubscriberDemo.SubscriberApplication : No active
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat in
main] o.apache.catalina.core.StandardService : Starting
    
```

Kiểm tra lại message broker

Overview				Messages			Message rates				+/-
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver	get	ack	
queue_1	classic	D Args	idle	0	0	0	0.00/s	0.00/s	0.00/s	0.00/s	
queue_2	classic	D Args	idle	0	0	0	0.00/s	0.00/s	0.00/s	0.00/s	
queue_3	classic	D Args	idle	0	0	0	0.00/s	0.00/s	0.00/s	0.00/s	

▼ Add a new queue

IV. Tài liệu tham khảo

https://grokonez.com/java-integration/distributed-system/rabbitmq-create-spring-rabbitmq-publishsubscribe-pattern-springboot#I_Spring_RabbitMq_PublishSubscribe_pattern
<https://www.rabbitmq.com/install-windows-manual.html>
<https://www.rabbitmq.com/which-erlang.html>

V. Các quy định khác