

Chapter 5

Working Queues and Routing Via RabbitMQ

บรรยายโดย ผศ.ดร.ธราวิเชษฐ์ ธิติจรรุญโรจน์ และอาจารย์สัญญาชัย น้อยจันทร์

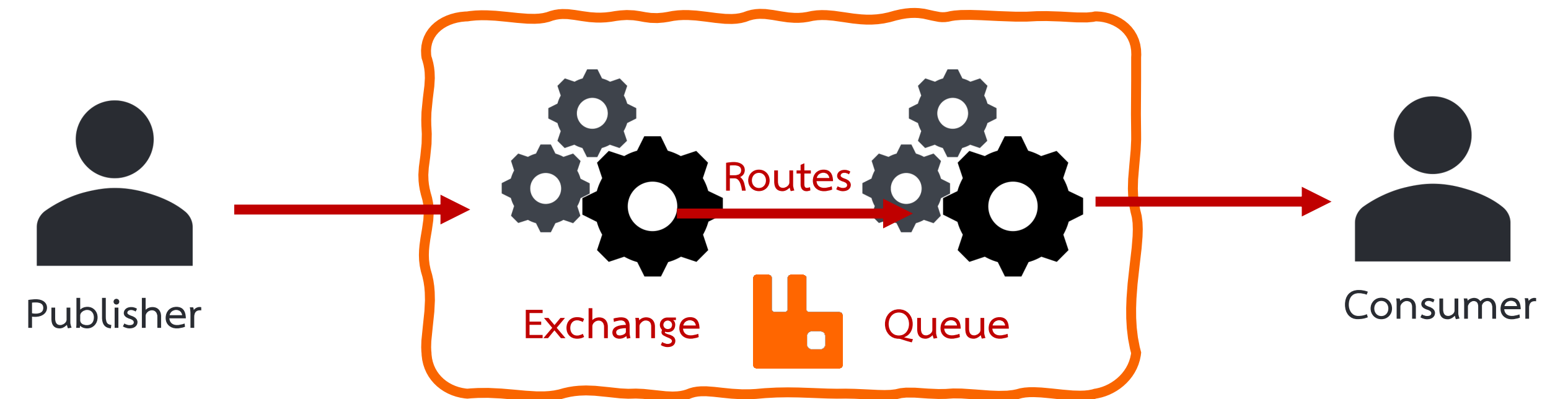
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



Outline

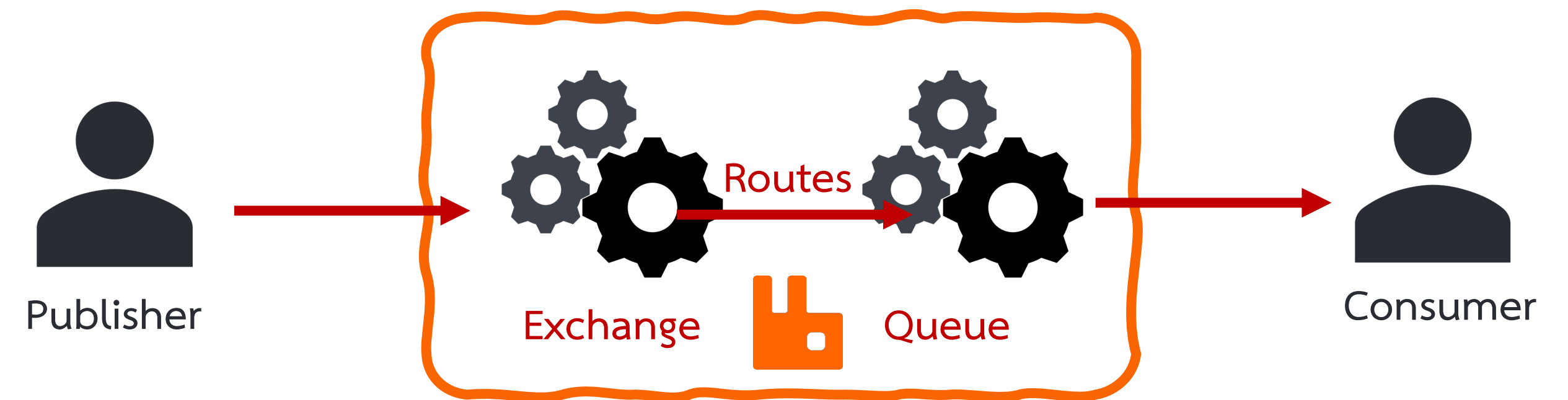
- Work queues
- Exchange type
 - Direct Exchange (default)
 - Fanout Exchange
 - Topic Exchange
 - Headers Exchange





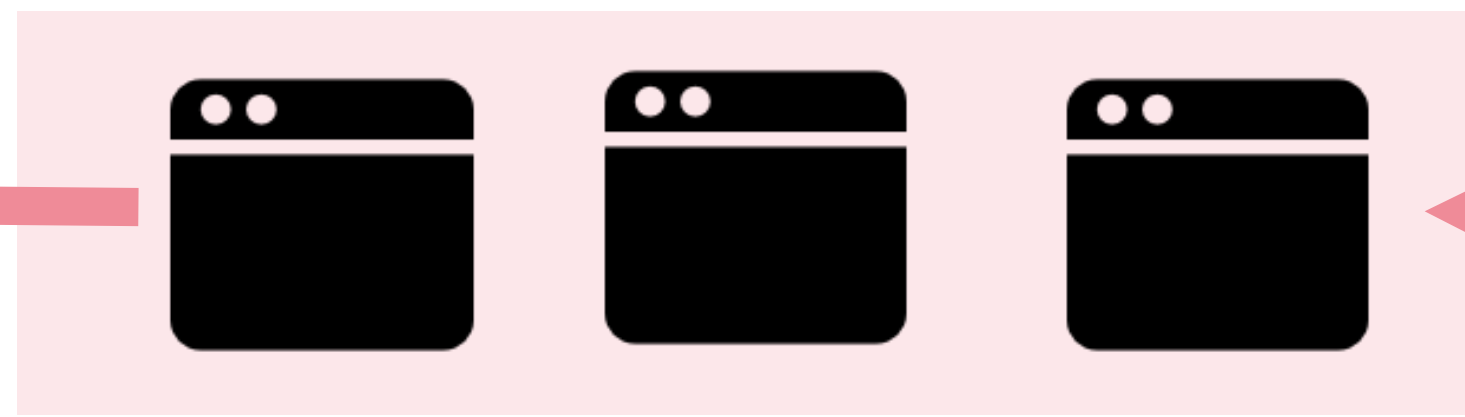
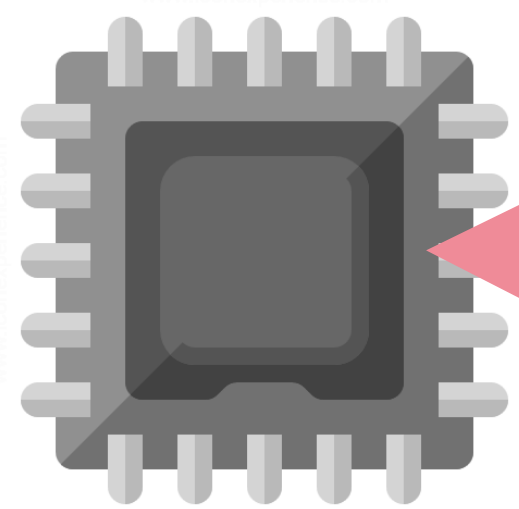
Outline

- Work queues
- Exchange type
 - Direct Exchange (default)
 - Fanout Exchange
 - Topic Exchange
 - Headers Exchange

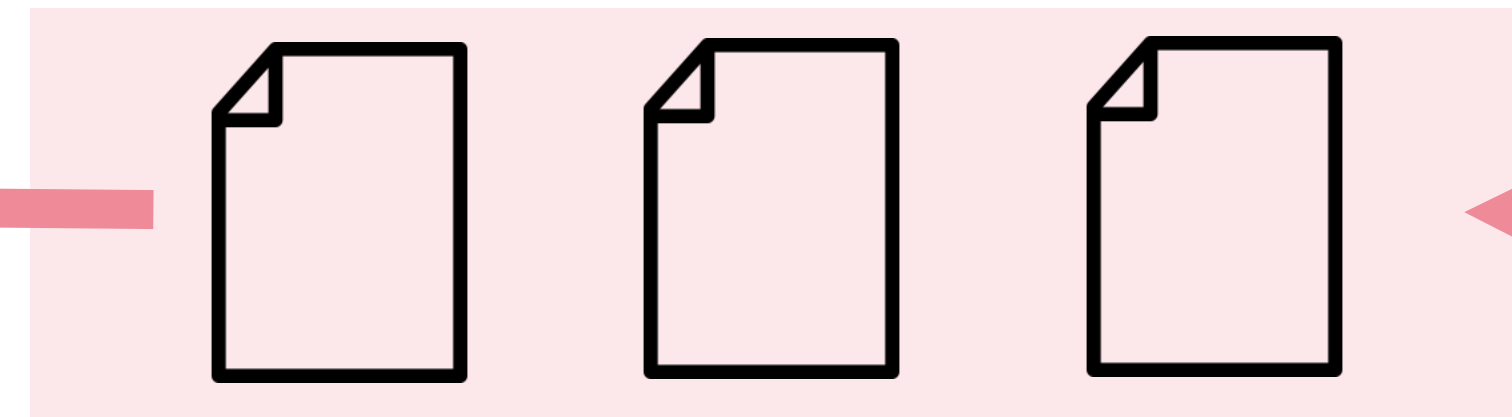




Queue คืออะไร



(ก) การเข้าแถวคอยของโปรเซสเพื่อรอให้หน่วยประมวลได้ประมวลผล

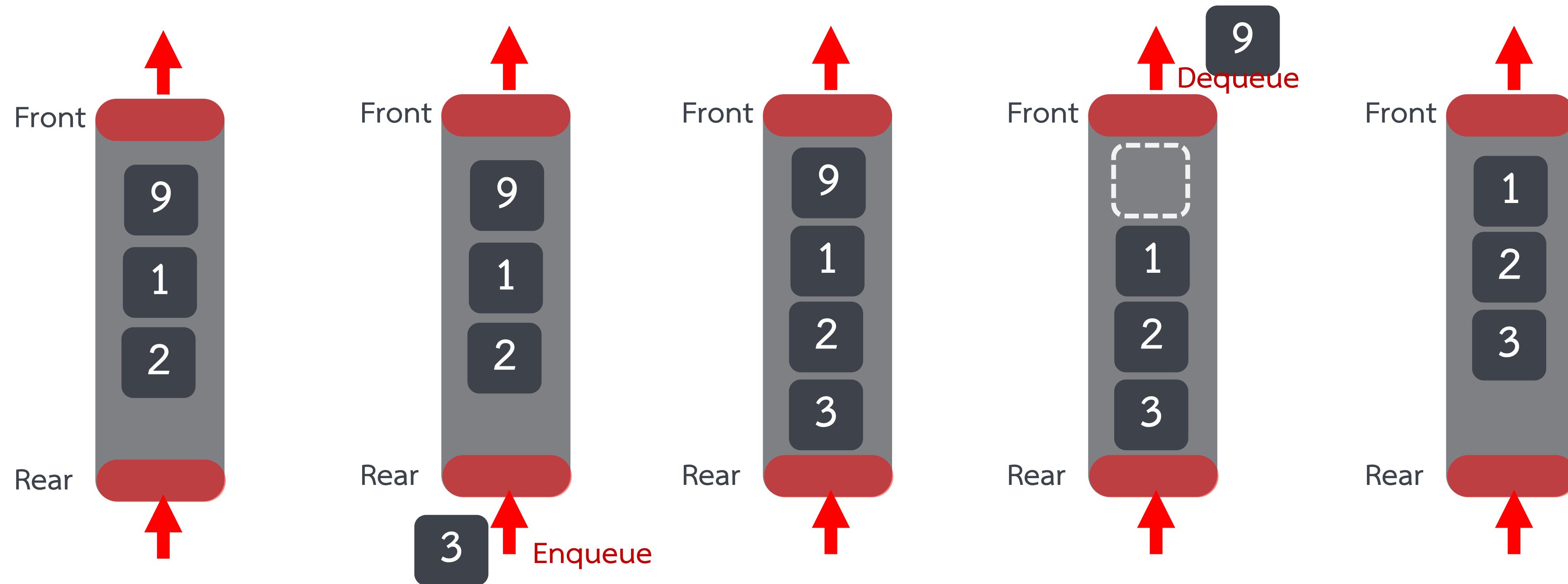


(ข) การเข้าแถวคอยของไฟล์เอกสารเพื่อรอพิมพ์ทางเครื่องพิมพ์

คือ โครงสร้างที่จัดเก็บข้อมูลต่อเนื่อง ซึ่งมีลักษณะคล้ายกับการเข้าแถวเพื่อรอรับบริการต่าง ๆ เป็นต้น คิวเป็นโครงสร้างข้อมูลที่ต้อง**กำหนดขนาดไว้ก่อน (แต่ใน RabbitMQ ไม่จำเป็น)** และจะเกิด “Overflow” ก็ต่อเมื่อคิวดังกล่าวเก็บข้อมูลเต็มขนาดที่กำหนดไว้ และถ้านำข้อมูลออกมาหมดแล้วแต่ยังมีการร้องขอข้อมูลอีกจะเกิด “Underflow”



Queue คืออะไร



คิวทำงานอยู่บนหลักการ “เข้าก่อนออกก่อน (First-In First-Out: FIFO)” โดยอาศัยตัวดำเนิน Enqueue สำหรับเพิ่ม Message ลงไปใน Queue ขณะที่ Dequeue สำหรับนำ Message ออกจาก Queue เพื่อส่งไปให้ Service ปลายทาง คิวถือว่ามีบทบาทสำคัญกับ Messaging Protocols กล่าวคือ Publisher และ Consumer จะสื่อสารกันผ่านคิวที่ทำหน้าที่คล้ายกับกลไกการจัดเก็บ (“queue-like storage mechanism”)



คุณสมบัติของ Queue คืออะไร



คือ สิ่งที่ใช้ระบุพฤติกรรมของคิวประกอบด้วย

- **ชื่อ (Name)** เป็นสิ่งสำคัญที่ Service หรือ Application ใช้สำหรับอ้างอิงถึงคิวดังกล่าว ซึ่งชื่อของคิวใน RabbitMQ สามารถยาวได้ถึง 255 Bytes ของ UTF-8 นอกจากนี้ สำหรับ RabbitMQ ชื่อคิวที่ขึ้นต้นด้วย “amp.” ถือว่าเป็นคำสงวนที่ใช้ภายใน RabbitMQ เท่านั้น
- **ความทนทาน (Durable)** กล่าวคือ จะให้คิวเก็บ Message ไว้หรือไม่ถ้า Broker เกิดการเริ่มการทำงานใหม่ (Restart)
- **เฉพาะตัว (Exclusive)** เป็นการกำหนดให้ใช้คิวได้เพียงครั้งเดียวต่อหนึ่งการเชื่อมต่อ หลังจากนั้น คิวจะถูกลบออกเมื่อการเชื่อมต่อปิดตัวลง
- **ลบอัตโนมัติ (Auto-delete)** เป็นการกำหนดให้ลบคิวทิ้งถ้าไม่เหลือ Consumer

อื่น ๆ อาทิเช่น message TTL และ queue length limit เป็นต้น อย่างไรก็ตาม การใช้คุณสมบัติทั้งหมดอาจจะไม่สมเหตุผลผลในการนำมาใช้งานจริง



การสั่งให้ RabbitMQ เริ่มการทำงาน

```
ibankii — beam.smp - rabbitmq-server — 103x38
Last login: Thu Oct 14 15:58:55 on ttys000
[ibankii@Mac-mini-khxng-Taravichet ~ % rabbitmq-server]
2021-10-14 16:08:34.257815+07:00 [info] <0.221.0> Feature flags: list of feature flags found:
2021-10-14 16:08:34.261522+07:00 [info] <0.221.0> Feature flags: [ ] implicit_default_bindings
2021-10-14 16:08:34.261545+07:00 [info] <0.221.0> Feature flags: [ ] maintenance_mode_status
2021-10-14 16:08:34.261562+07:00 [info] <0.221.0> Feature flags: [ ] quorum_queue
2021-10-14 16:08:34.261573+07:00 [info] <0.221.0> Feature flags: [ ] stream_queue
2021-10-14 16:08:34.261604+07:00 [info] <0.221.0> Feature flags: [ ] user_limits
2021-10-14 16:08:34.261615+07:00 [info] <0.221.0> Feature flags: [ ] virtual_host_metadata
2021-10-14 16:08:34.261625+07:00 [info] <0.221.0> Feature flags: feature flag states written to disk: yes
2021-10-14 16:08:34.371073+07:00 [noti] <0.44.0> Application syslog exited with reason: stopped
2021-10-14 16:08:34.371122+07:00 [noti] <0.221.0> Logging: switching to configured handler(s); following messages may not be visible in this log output

## ##      RabbitMQ 3.9.7
## ##
##### Copyright (c) 2007-2021 VMware, Inc. or its affiliates.
#####
##### Licensed under the MPL 2.0. Website: https://rabbitmq.com

Erlang:      24.1.2 [emu]
TLS Library: OpenSSL - OpenSSL 1.1.1l  24 Aug 2021

Doc guides:  https://rabbitmq.com/documentation.html
Support:     https://rabbitmq.com/contact.html
Tutorials:   https://rabbitmq.com/getstarted.html
Monitoring:  https://rabbitmq.com/monitoring.html

Logs: /opt/homebrew/var/log/rabbitmq/rabbit@localhost.log
      /opt/homebrew/var/log/rabbitmq/rabbit@localhost_upgrade.log
      <stdout>

Config file(s): (none)
Starting broker... completed with 7 plugins.
```

RabbitMQ พร้อมทำงาน

เปิด Terminal และพิมพ์คำสั่ง
“rabbitmq-server” เพื่อให้ RabbitMQ เริ่ม
การทำงาน จากนั้น รอ RabbitMQ เริ่มการ
ทำงาน โดยสังเกตที่ Terminal ว่าปรากฏ
ข้อความต่อไปนี้ “Starting broker ...
completed with 7 plugins” แสดงว่า
RabbitMQ พร้อมให้บริการ



การสร้าง Queue ใน RabbitMQ

ขั้นที่ 1 เปิด Browser เพื่อเข้าสู่หน้าจัดการ RabbitMQ ดังภาพต่อไปนี้

http://localhost:15672/

RabbitMQ Erlang Version Requirements — Rab... RabbitMQ Management

RabbitMQ™

Username: *


Password: *

Login



การสร้าง Queue ใน RabbitMQ

ขั้นที่ 2 หน้า RabbitMQ จะปรากฏขึ้นดังภาพ จากนั้น เลือกเมนู Queues ด้านบนดังภาพต่อไปนี้

 **RabbitMQ**TM

RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-14 16:16:21 Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest Log out

Overview

Connections

Channels

Exchanges

Queues

Admin

Overview

Totals

Queued messages last minute ?

Currently idle

Message rates last minute ?

Currently idle

Global counts ?

Connections: 0 Channels: 0 Exchanges: 7 Queues: 0 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@localhost	58 256 available	0 141 available	436 1048576 available	91 MiB 6.4 GiB high watermark	284 GiB 48 MiB low watermark	7m 42s	basic disc 5 rss	This node All nodes	



การสร้าง Queue ใน RabbitMQ

ขั้นที่ 3 เลือกเมนู “Add a new queue” ดังภาพต่อไปนี้

RabbitMQ™

RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-14 16:16:27 Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest Log out

Overview

Connections

Channels

Exchanges

Queues

Admin

Queues

▼ All queues (0)

Pagination

Page 1 of 0 - Filter: ☐ Regex ?

Displaying 0 item , page size up to: 100

... no queues ...

► Add a new queue

HTTP API

Server Docs

Tutorials

Community Support

Community Slack

Commercial Support

Plugins

GitHub

Changelog



การสร้าง Queue ใน RabbitMQ

ขั้นที่ 4 ตั้งชื่อ Queue บริเวณ Name จากนั้น กด “Add queue” ดังภาพต่อไปนี้

... no queues ...

▼ Add a new queue

Type: Classic

Name: Queue1 *

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



การสร้าง Queue ใน RabbitMQ

ขั้นที่ 5 ถ้าสร้างสำเร็จจะปรากฏผลลัพธ์ดังภาพต่อไปนี้

 RabbitMQ™

RabbitMQ 3.9.7

Erlang 24.1.2

Refreshed 2021-10-14 16:18:22

Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest

Log out

Overview

Connections

Channels

Exchanges

Queues

Admin

All queues (1)

Pagination

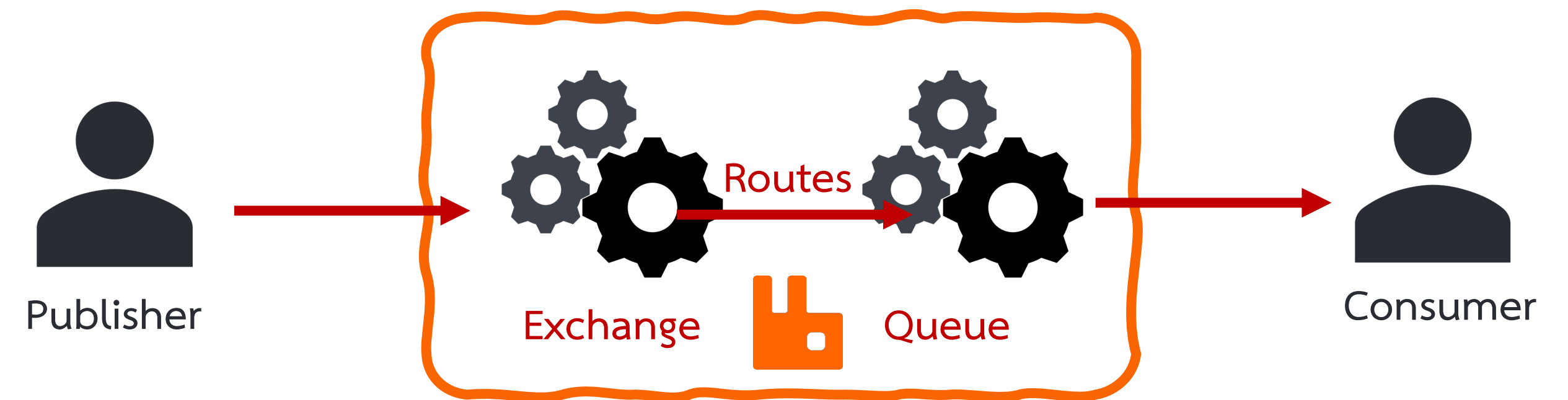
Page 1 of 1 - Filter: ☐ Regex ?

Displaying 1 item , page size up to: 100



Outline

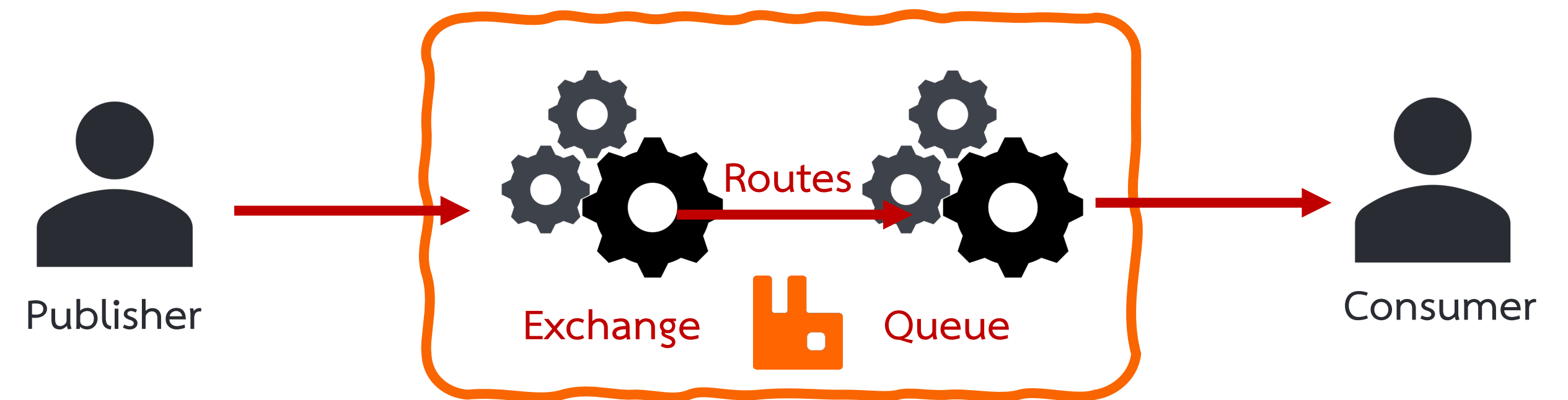
- Work queues
- Exchange type
 - Direct Exchange (default)
 - Fanout Exchange
 - Topic Exchange
 - Headers Exchange

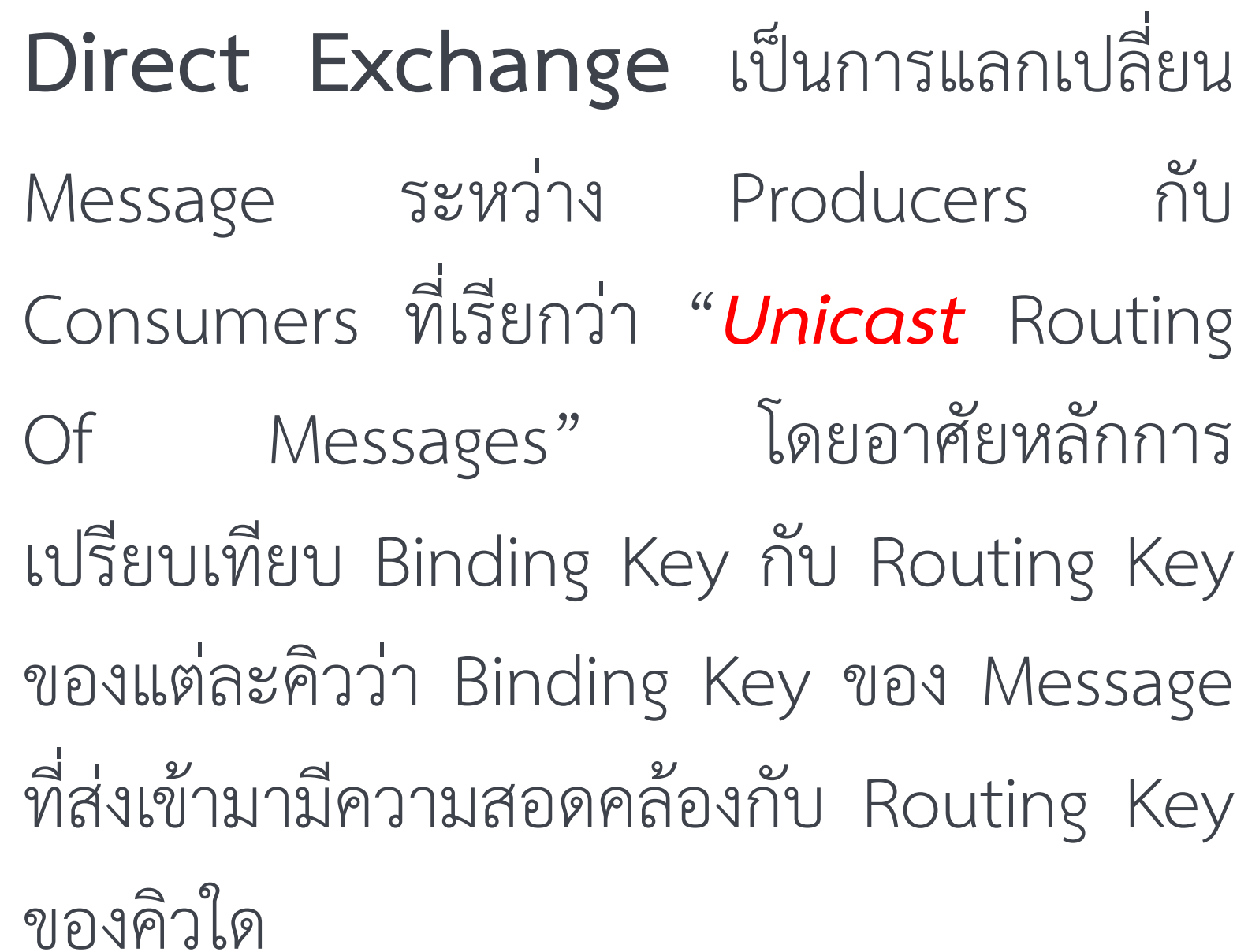




Outline

- Work queues
- Exchange type
 - Direct Exchange (default)
 - Fanout Exchange
 - Topic Exchange
 - Headers Exchange







กำหนดค่า RabbitMQ แบบ Direct Exchange

ขั้นที่ 1 สร้างคิวชื่อ Computer, Mobile และ TV

Overview Connections **Channels** Exchanges **Queues** Admin

Queues

▼ All queues (1)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Overview				Messages			Message rates			+/-
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
Queue1	classic	D Args	idle	0	0	0				

▼ Add a new queue

Type: Classic

Name: *

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

(1) คลิกเลือกเมนู "Queues"

(2) กำหนดชื่อคิว

(3) กดสร้างคิว

(4) ผลลัพธ์หลังการสร้างทั้ง 3 คิว

Overview			
Name	Type	Features	State
Computer	classic	D Args	idle
Mobile	classic	D Args	idle
TV	classic	D Args	idle



กำหนดค่า RabbitMQ แบบ Direct Exchange

ขั้นที่ 2 สร้าง Exchanges

RabbitMQ™

RabbitMQ 3.9.7

Erlang 24.1.2

Refreshed 2021-10-18 14:31:05

Refresh every 5 seconds

Virtual host

All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest

Log out

Overview

Connections

Channels

Exchanges

Queues

Admin

Exchanges

▼ All exchanges (7)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Displaying 7 items , page size up to: 100

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

(2)เลือกเพื่อสร้างรูปแบบการ Exchange

HTTP API

Server Docs

Tutorials

Community Support

Community Slack

Commercial Support

Plugins

GitHub

Changelog



กำหนดค่า RabbitMQ แบบ Direct Exchange

ขั้นที่ 2 สร้าง Exchanges

Refreshed 2021-10-18 14:31:05 Refresh every 5 seconds Virtual host All Cluster rabbit@Mac-mini-khxng-Taravichet User guest Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

▼ All exchanges (7)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

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	I			
amq.topic	topic	D			

► Add a new exchange

▼ Add a new exchange

Name: *

Type: ▼

Durability: ▼

Auto delete: ? ▼

Internal: ? ▼

Arguments: String ▼

Add Alternate exchange ?

Add exchange

(1) คลิกเลือกเมนู “Exchanges”

(3) ระบุชื่อ “Exchanges”

(4) เลือกชนิด “direct”

(5) กดปุ่ม “Add exchange”

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog



กำหนดค่า RabbitMQ แบบ Direct Exchange

ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

RabbitMQ™

RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-18 14:35:47 Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest Log out

OverviewConnectionsChannelsExchangesQueuesAdmin

Exchanges

▼ All exchanges (8)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Displaying 8 items , page size up to: 100

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
MyDirectExchange	direct	D			
amq.direct	direct	D			
amq.fanout	fanout	F			

คลิกที่ชื่อ Exchange เพื่อกำหนดการเชื่อมโยง



กำหนดค่า RabbitMQ แบบ Direct Exchange

ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

Name	Type	Features	Message rate
(AMQP default)	direct	D	
MyDirectExchange	direct	D	
amq.direct	direct	D	
amq.fanout	fanout	F	

Exchange: MyDirectExchange

Overview

Bindings

This exchange

↓

... no bindings ...

Add binding from this exchange

To queue: * (1) ระบุชื่อ "Queue" ที่ต้องการ Route กับ Exchange

Routing key: (2) ระบุชื่อ "Routing Key"

Arguments: = String

Bind (3) กด "Bind" เพื่อ Route

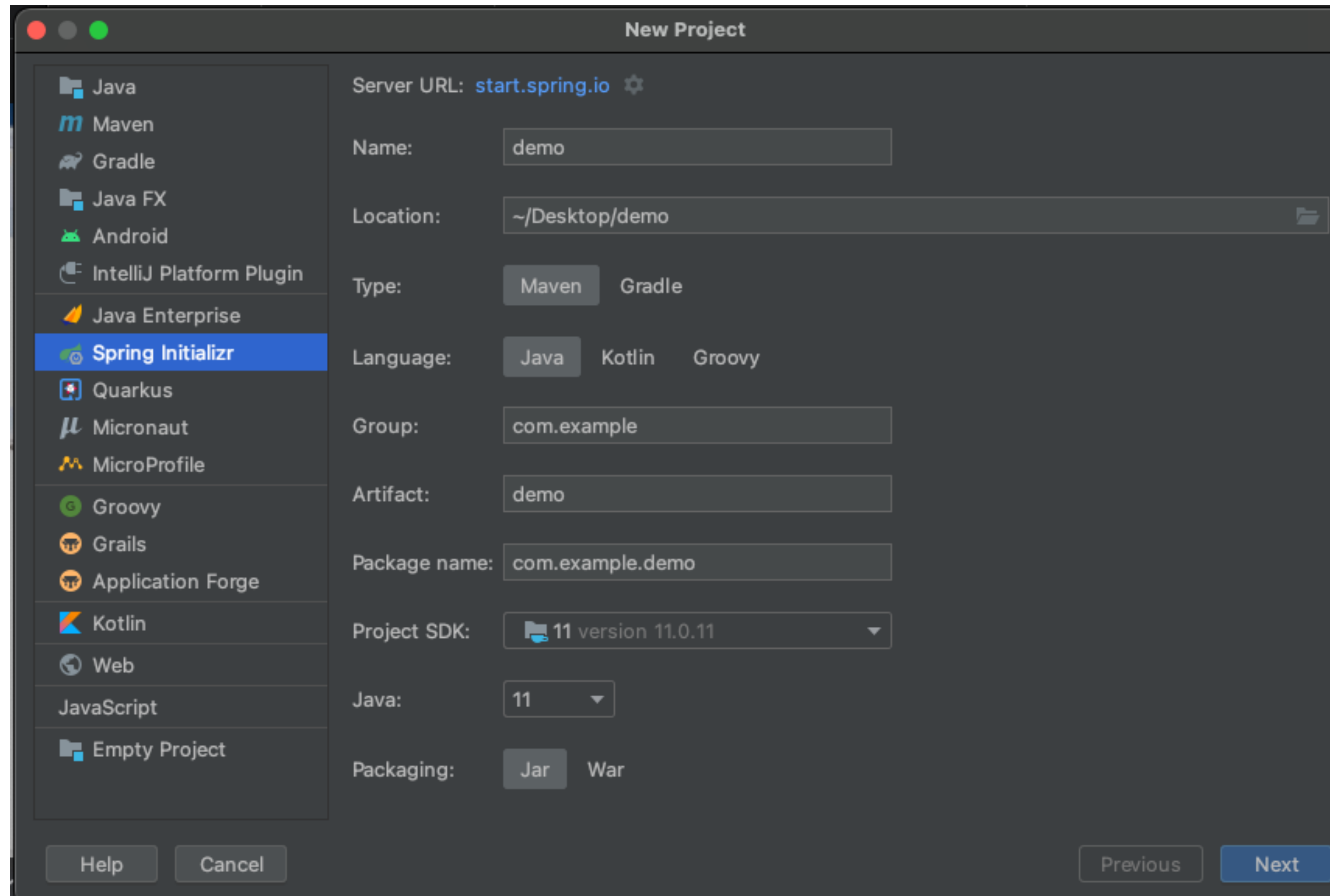


The screenshot displays the RabbitMQ web interface. On the left, the 'Exchanges' sidebar shows a list of exchanges: '(AMQP default)', 'MyDirectExchange', 'amq.direct', and 'amq.fanout'. 'MyDirectExchange' is highlighted with a red box. A red dashed line connects this box to a detailed view of the exchange's bindings. This view shows a table with columns 'To', 'Routing key', 'Arguments', and 'Unbind'. The table lists four bindings: 'Computer' with routing key 'com', 'Computer' with 'notebook', 'Mobile' with 'mobile', and 'TV' with 'tv'. Below the table, there is a form to 'Add binding from this exchange' with fields for 'To queue', 'Routing key', and 'Arguments', and a 'Bind' button.



Spring Boot กับ RabbitMQ แบบ Direct Exchange

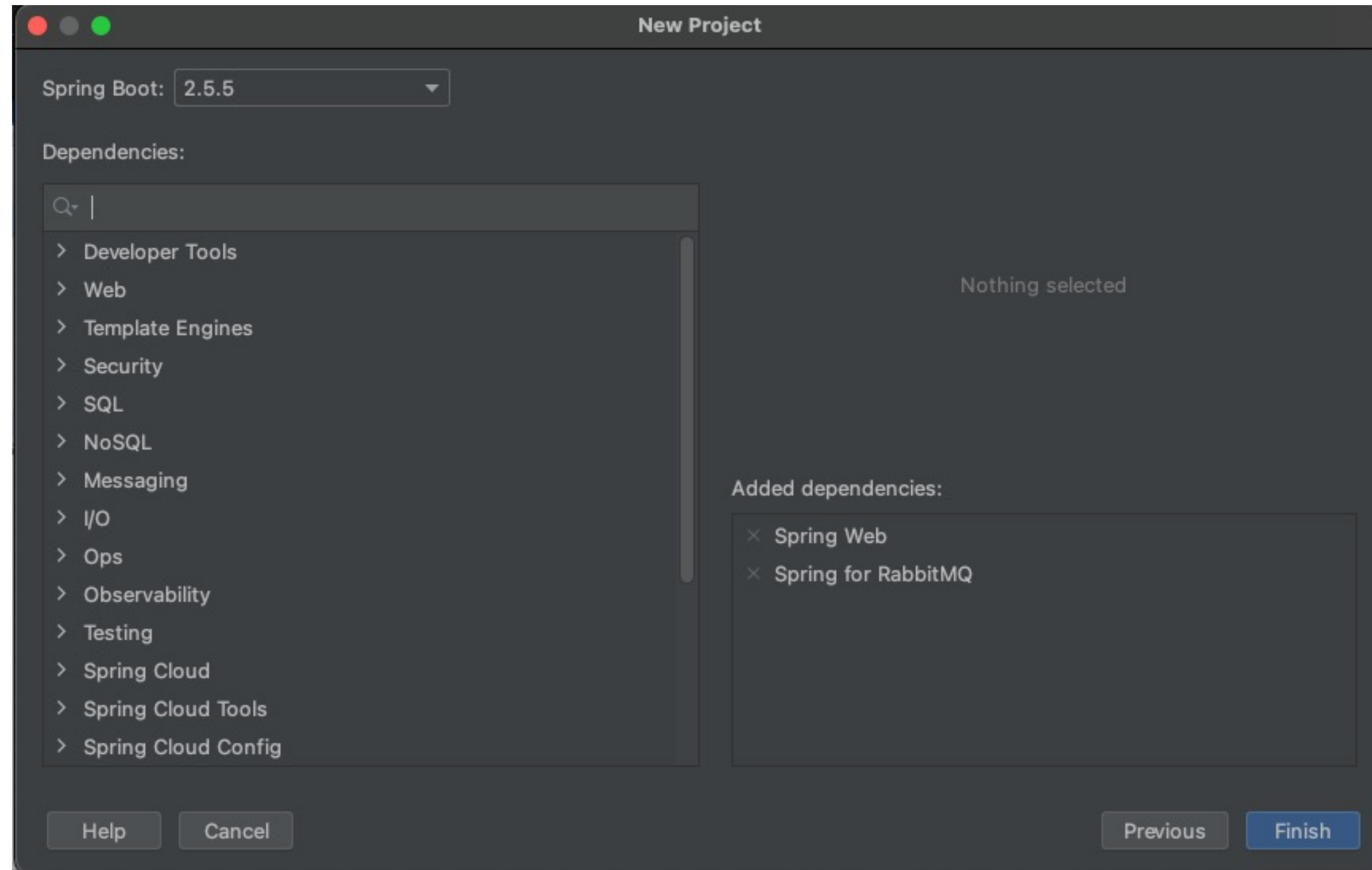
ขั้นที่ 1 สร้างโปรเจกต์ด้วย Spring Initializr ร่วมกับเฟรมเวิร์ก Maven





Spring Boot กับ RabbitMQ แบบ Direct Exchange

ขั้นที่ 2 เลือก Dependencies 2 ตัว ได้แก่ (1) Spring Web และ (2) Spring for RabbitMQ





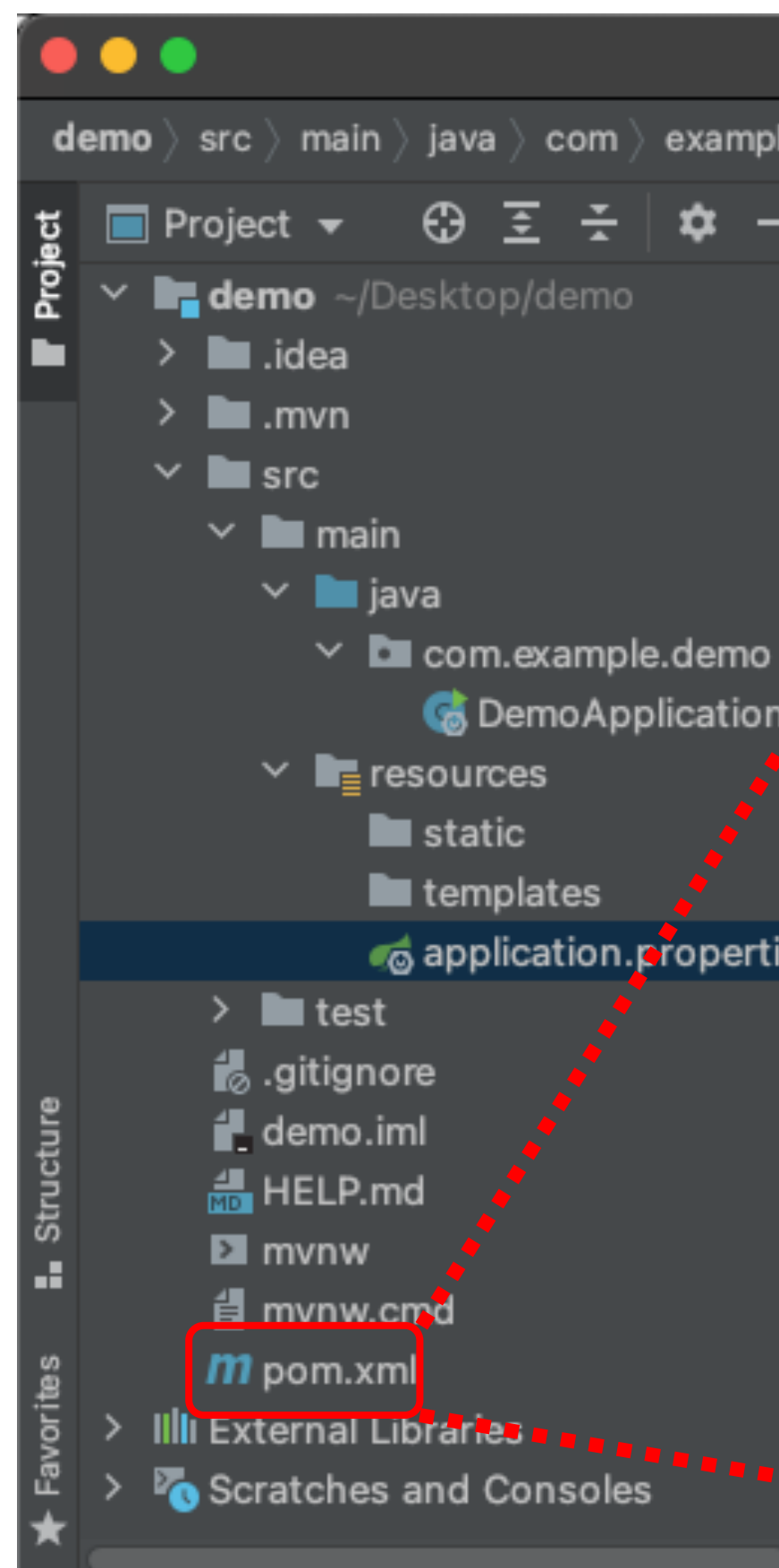
Spring Boot กับ RabbitMQ แบบ Direct Exchange

```
demo - DemoApplication.java  
demo > src > main > java > com > example > demo > DemoApplication  
Project  
demo ~/Desktop/demo  
  .idea  
  .mvn  
  src  
    main  
      java  
        com.example.demo  
          DemoApplication  
      resources  
        static  
        templates  
    application.properties  
    test  
  .gitignore  
  demo.iml  
  HELP.md  
  mvnw  
  mvnw.cmd  
  pom.xml  
Favorites  
  External Libraries  
  Scratches and Consoles  
Database  
Maven  
1 package com.example.demo;  
2  
3 import org.springframework.boot.SpringApplication;  
4 import org.springframework.boot.autoconfigure.SpringBootApplication;  
5  
6 @SpringBootApplication  
7 public class DemoApplication {  
8  
9     public static void main(String[] args) { SpringApplication.run(DemoApplication.class, args); }  
10  
11  
12 }  
13  
14  
2 Event Log  
File pattern '*.apk' (from 'bundled' plugin) was reassigned to file type 'APK' by 'Android' plugin: You can c... (6 minutes ago) Downloading patch files 2:1 LF UTF-8 4 spaces
```




Spring Boot กับ RabbitMQ แบบ Direct Exchange

ขั้นที่ 3 ตรวจสอบ Dependencies ในไฟล์ pom.xml

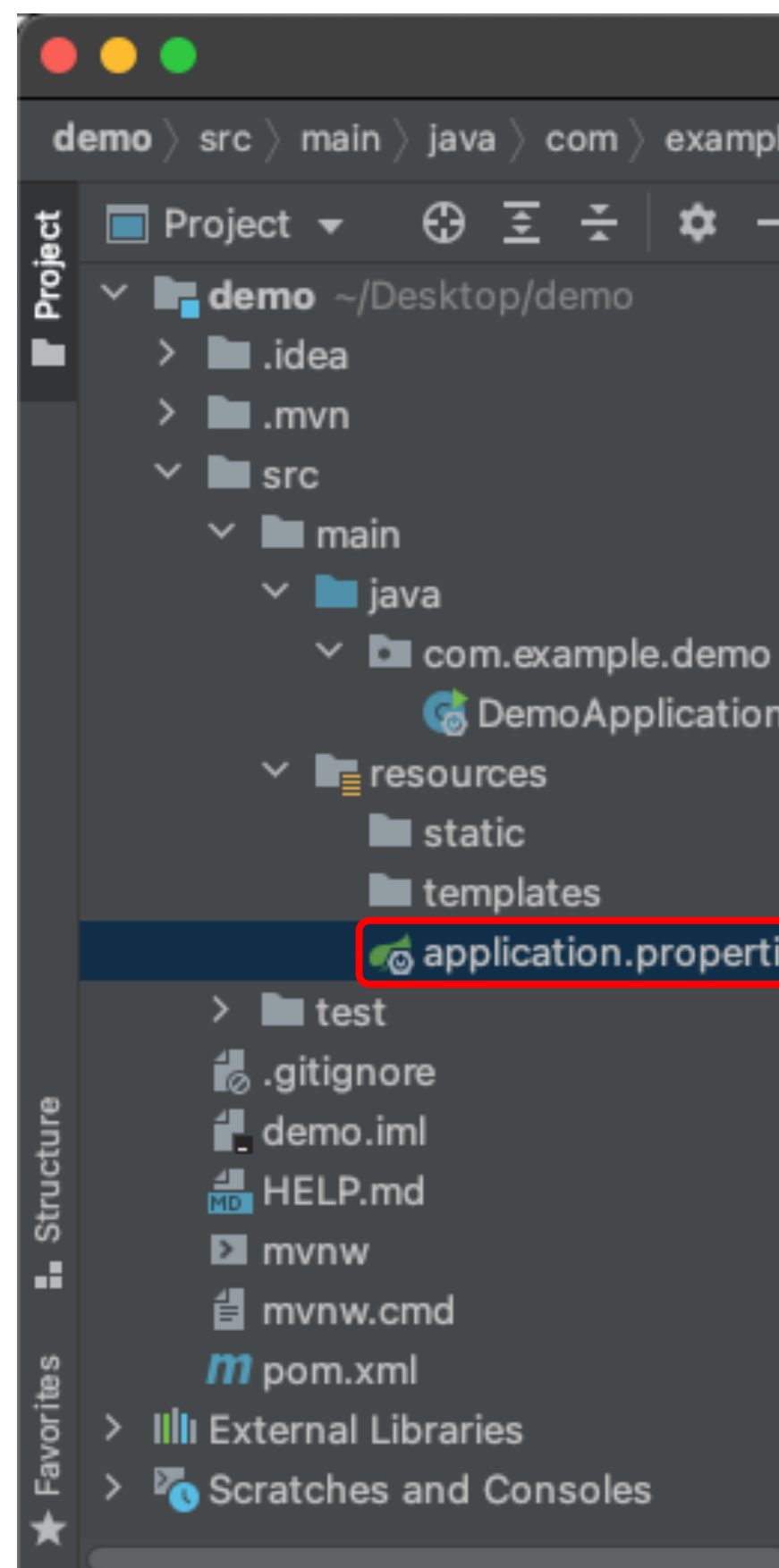


```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-amqp</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.amqp</groupId>
    <artifactId>spring-rabbit-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```



Spring Boot กับ RabbitMQ แบบ Direct Exchange

ขั้นที่ 4 กำหนดค่า Config พื้นฐานให้ RabbitMQ กับ Spring Boot ในไฟล์ application.properties



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



```
import org.springframework.amqp.rabbit.core.RabbitTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
public class Publisher01 {
    @Autowired // แจ้ง Spring ว่าได้มีประกาศแอททริบิวต์ของคลาสขึ้นมา ช่วยกำหนด Object ที่มีชนิด (type/sub-type) เป็น RabbitTemplate มาใส่หรือสร้างใหม่ให้ด้วย
    private RabbitTemplate rabbitTemplate;

    @RequestMapping(value = "/pow2/{num}", method = RequestMethod.GET)
    public String pow2(@PathVariable("num") int num) {
        System.out.println("Publisher In : " + num );

        rabbitTemplate.convertAndSend("MyDirectExchange", "mobile", num);
        //      Exchange Name      Routing Key      Value
        //      -----

        return "Success";
    }
}
```



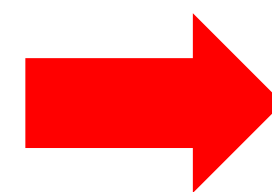
Spring Boot กับ RabbitMQ แบบ Direct Exchange

การสร้าง **Consumer** ให้ฟัง Request Message จาก Queue โดยอาศัย RabbitListener

```
import org.springframework.amqp.rabbit.annotation.RabbitListener;
import org.springframework.stereotype.Service;

@Service
public class Consumer01 {

    Queue Name
    @RabbitListener(queues = "Mobile")
    public void pow2Consumer(int num){
        System.out.println("pow2Consumer : "+ (num*num) );
    }
}
```



```
Publisher In : 4
pow2Consumer : 16
```




Spring Boot กับ RabbitMQ แบบ Direct Exchange

การส่งข้อมูลที่เป็น Object ที่เราสร้างมาเป็น Message ต้อง implements อินเตอร์เฟส Serializable เสมอ

```
@RestController
public class Publisher02 {
    @Autowired
    private RabbitTemplate rabbitTemplate;
    @RequestMapping(value = "/pow3/{num}", method = RequestMethod.GET)
    public String pow2(@PathVariable("num") int num){
        NumMessage n = new NumMessage(num);
        System.out.println("Publisher In : " + n.getNum() );
        rabbitTemplate.convertAndSend( exchange: "MyDirectExchange", routingKey: "mobile",n);
        return "Success";
    }
}
```

```
@Service
public class Consumer02 {
    @RabbitListener(queues = "Mobile")
    public void pow2Consumer(NumMessage num){
        int n = num.getNum();
        System.out.println("pow2Consumer : "+ (n*n));
    }
}
```

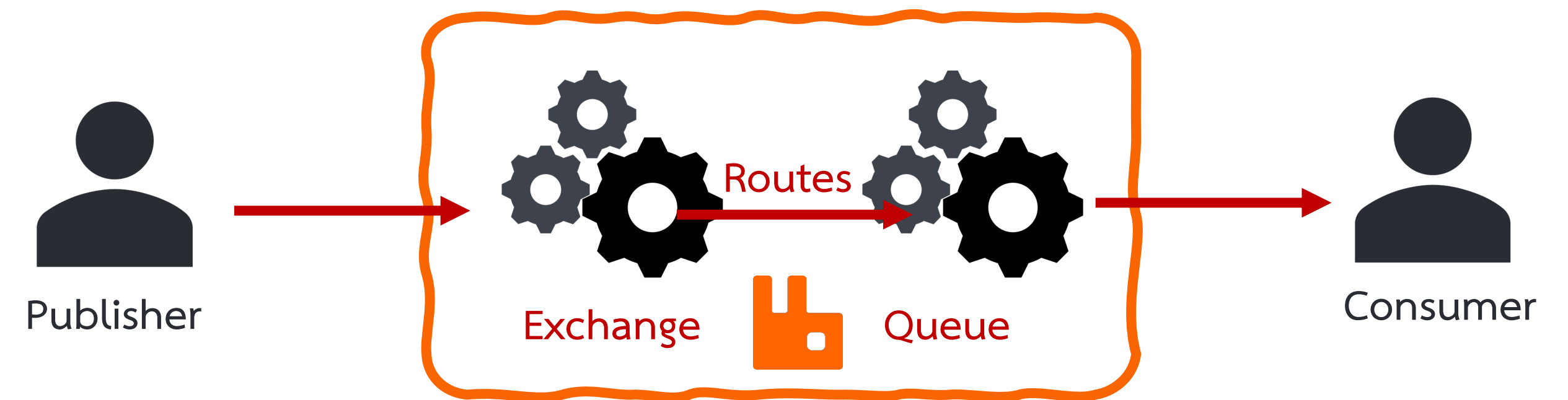
```
public class NumMessage implements Serializable {
    private int num;
    public void setNum(int num) { this.num = num; }
    public int getNum() { return this.num; }

    public NumMessage(){
    }
    public NumMessage(int num){ this.setNum(num);}
}
```



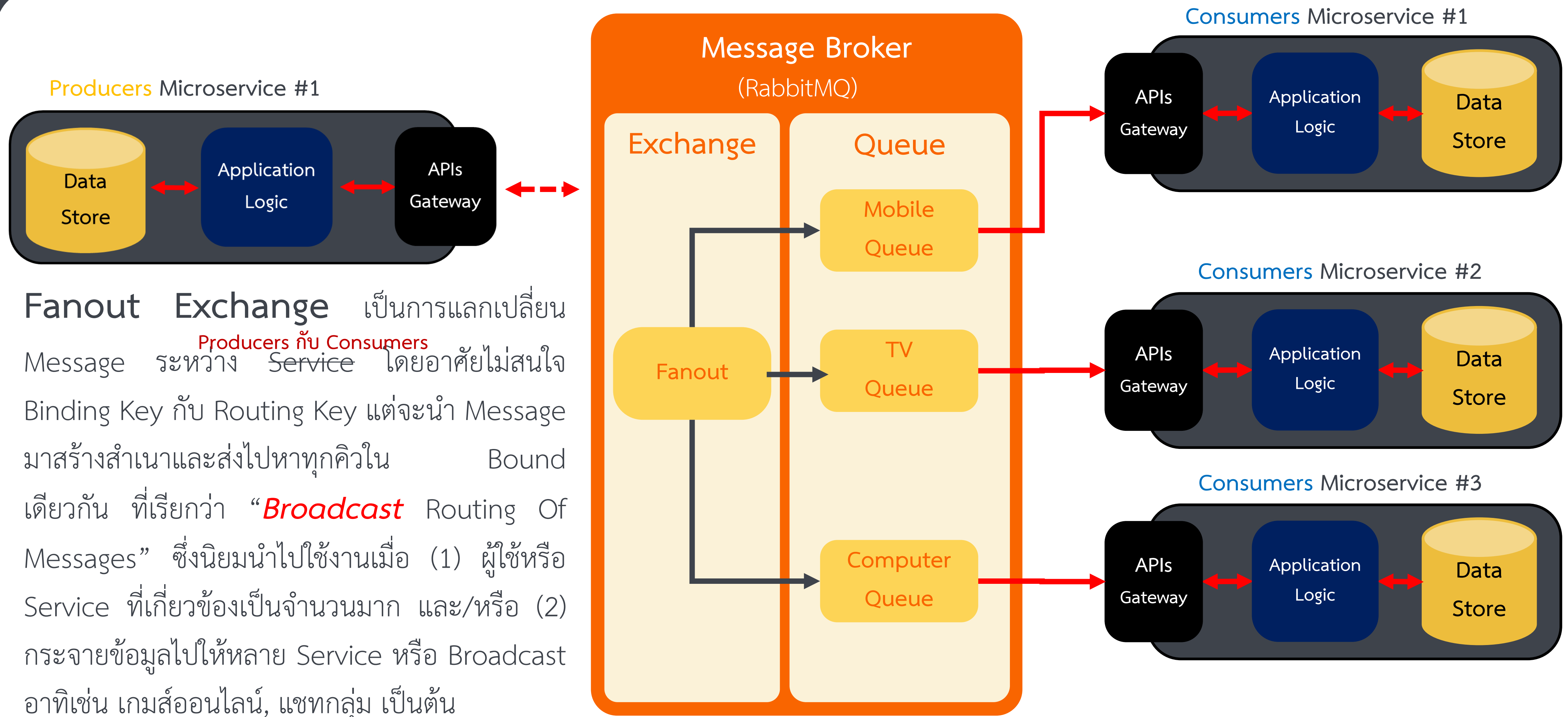
Outline

- Work queues
- Exchange type
 - Direct Exchange (default)
 - Fanout Exchange
 - Topic Exchange
 - Headers Exchange





Fanout Exchange



Fanout Exchange เป็นการแลกเปลี่ยน Message ระหว่าง **Producers** กับ **Consumers** โดยอาศัยไม่สนใจ Binding Key กับ Routing Key แต่จะนำ Message มาสร้างสำเนาและส่งไปหาทุกคิวใน Bound เดียวกัน ที่เรียกว่า “**Broadcast** Routing Of Messages” ซึ่งนิยมนำไปใช้งานเมื่อ (1) ผู้ใช้หรือ Service ที่เกี่ยวข้องเป็นจำนวนมาก และ/หรือ (2) กระจายข้อมูลไปให้หลาย Service หรือ Broadcast อาทิเช่น เกมออนไลน์, แชทกลุ่ม เป็นต้น



กำหนดค่า RabbitMQ แบบ Fanout Exchange

ขั้นที่ 1 สร้างคิวชื่อ Computer, Mobile และ TV (ข้ามได้ถ้ามีคิวอยู่แล้ว)

Overview Connections **Channels** Exchanges **Queues** Admin

Queues

▼ All queues (1)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Overview				Messages			Message rates			
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	+/-
Queue1	classic	D Args	idle	0	0	0				

▼ Add a new queue

Type: Classic

Name: *

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

(1) คลิกเลือกเมนู "Queues"

(2) กำหนดชื่อคิว

(3) กดสร้างคิว

(4) ผลลัพธ์หลังการสร้างทั้ง 3 คิว

Overview				
Name	Type	Features	State	
Computer	classic	D Args	idle	
Mobile	classic	D Args	idle	
Queue1	classic	D Args	idle	
TV	classic	D Args	idle	



กำหนดค่า RabbitMQ แบบ Fanout Exchange

ขั้นที่ 2 สร้าง Exchanges

RabbitMQ™

RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-18 14:31:05 Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

▼ All exchanges (7)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Displaying 7 items , page size up to: 100

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 (2)เลือกเพื่อสร้างรูปแบบการ Exchange

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog



กำหนดค่า RabbitMQ แบบ Fanout Exchange

ขั้นที่ 2 สร้าง Exchanges

RabbitMQ™ RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-18 14:31:05 Refresh every 5 seconds Virtual host All Cluster rabbit@Mac-mini-khxng-Taravichet User guest Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

▼ All exchanges (7)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

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	I D			
amq.topic	topic	D			

► Add a new exchange

▼ Add a new exchange

Name: (3) ระบุชื่อ "Exchanges"

Type: (4) เลือกชนิด "fanout"

Durability:

Auto delete: ?

Internal: ?

Arguments: =

Add Alternate exchange ?

Add exchange (5) กดปุ่ม "Add exchange"

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog



กำหนดค่า RabbitMQ แบบ Fanout Exchange

ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

 **RabbitMQ**TM RabbitMQ 3.9.7 Erlang 24.1.2

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

► All exchanges (9)

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
MyDirectExchange	direct	D	0.00/s	0.00/s	
MyFanoutExchange	fanout	D			



กำหนดค่า RabbitMQ แบบ Fanout Exchange

ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

RabbitMQ™

RabbitMQ 3.9.7

OverviewConnectionsChannelsExchanges

Exchanges

All exchanges (8)

Pagination

Page 1 of 1 - Filter:

Name	Type	Features	Message rate
(AMQP default)	direct	D	
MyFanoutExchange	fanout	D	
amq.direct	direct	D	
amq.fanout	fanout	D	

Exchange: MyFanoutExchange

Overview

Bindings

This exchange

To	Routing key	Arguments	
Computer			Unbind
Mobile			Unbind
TV			Unbind

Add binding from this exchange

To queue *: (1) ระบุชื่อ "Queue"

Routing key: (2) ไม่ต้องระบุชื่อ "Routing Key"

Arguments: = String

Bind (3) กด "Bind" เพื่อ Route

4:35:47 Refresh every 5 seconds

Virtual host All

bbit@Mac-mini-khxng-Taravichet

User guest Log out

8 items , page size up to: 100



Spring Boot กับ RabbitMQ แบบ Fanout Exchange

การสร้าง **Publisher** ให้ส่ง Message ไปที่ Broker (RabbitMQ) โดยอาศัยคลาส RabbitTemplate

```
package com.example.demo;
import org.springframework.amqp.rabbit.core.RabbitTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
public class Publisher03 {

    @Autowired
    private RabbitTemplate rabbitTemplate;

    @RequestMapping(value = "/queueUp/{name}", method = RequestMethod.GET)
    public String queueUp(@PathVariable("name") String name){
        System.out.println("Publisher In : " + name );
        rabbitTemplate.convertAndSend("MyFanoutExchange", "", name);
        return "Success";
    }
}
```



Spring Boot กับ RabbitMQ แบบ Fanout Exchange

การสร้าง **Consumer** ให้ฟัง Request Message จาก Queue โดยอาศัย RabbitListener

```
package com.example.demo;
import org.springframework.amqp.rabbit.annotation.RabbitListener;
import org.springframework.stereotype.Service;

@Service
public class Consumer03 {

    @RabbitListener(queues = "TV")
    public void queueTV(String name){    System.out.println("In TV Queue : "+ name);    }

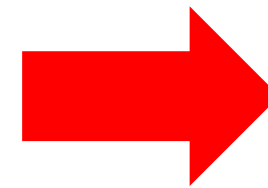
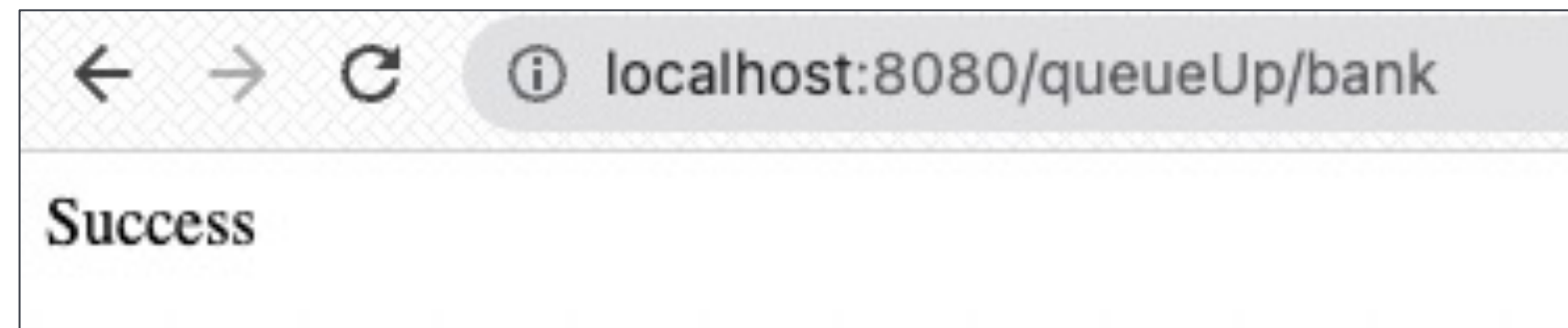
    @RabbitListener(queues = "Computer")
    public void queueCom(String name){    System.out.println("In Computer Queue : "+ name);    }

    @RabbitListener(queues = "Mobile")
    public void queueMobile(String name){    System.out.println("In Mobile Queue : "+ name);    }

}
```



Spring Boot กับ RabbitMQ แบบ Fanout Exchange

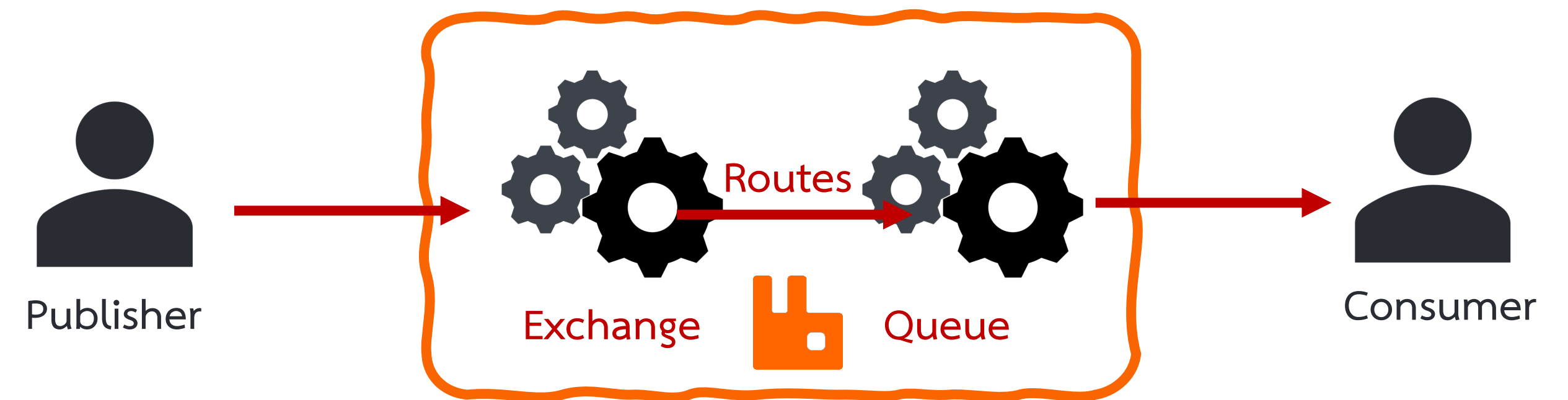


```
Publisher In : bank  
In Computer Queue : bank  
In TV Queue : bank  
In Mobile Queue : bank
```



Outline

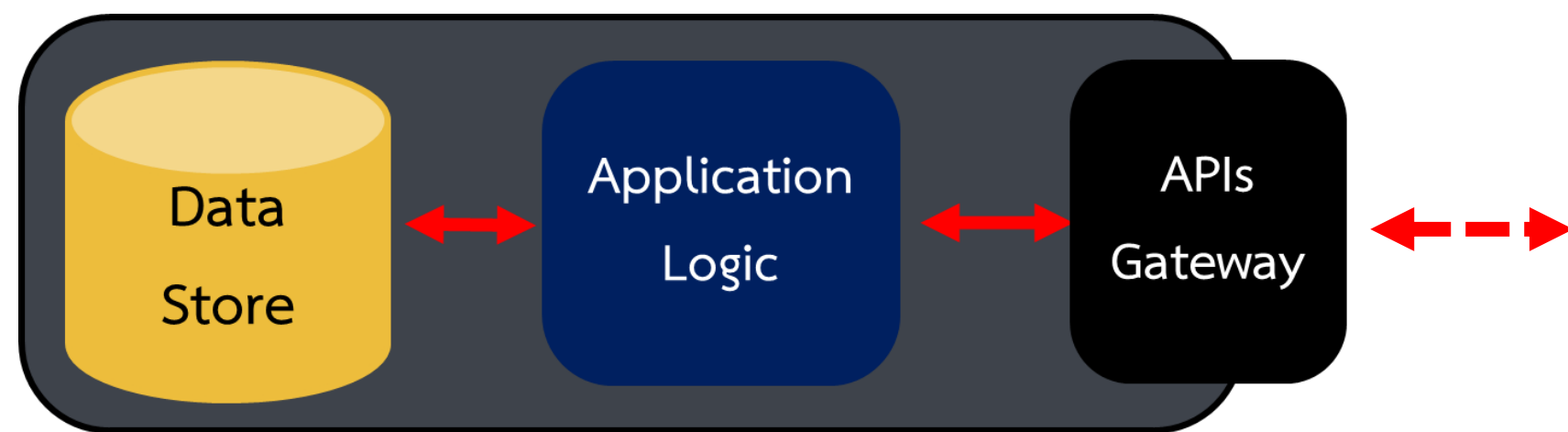
- Work queues
- Exchange type
 - Direct Exchange (default)
 - Fanout Exchange
 - Topic Exchange
 - Headers Exchange



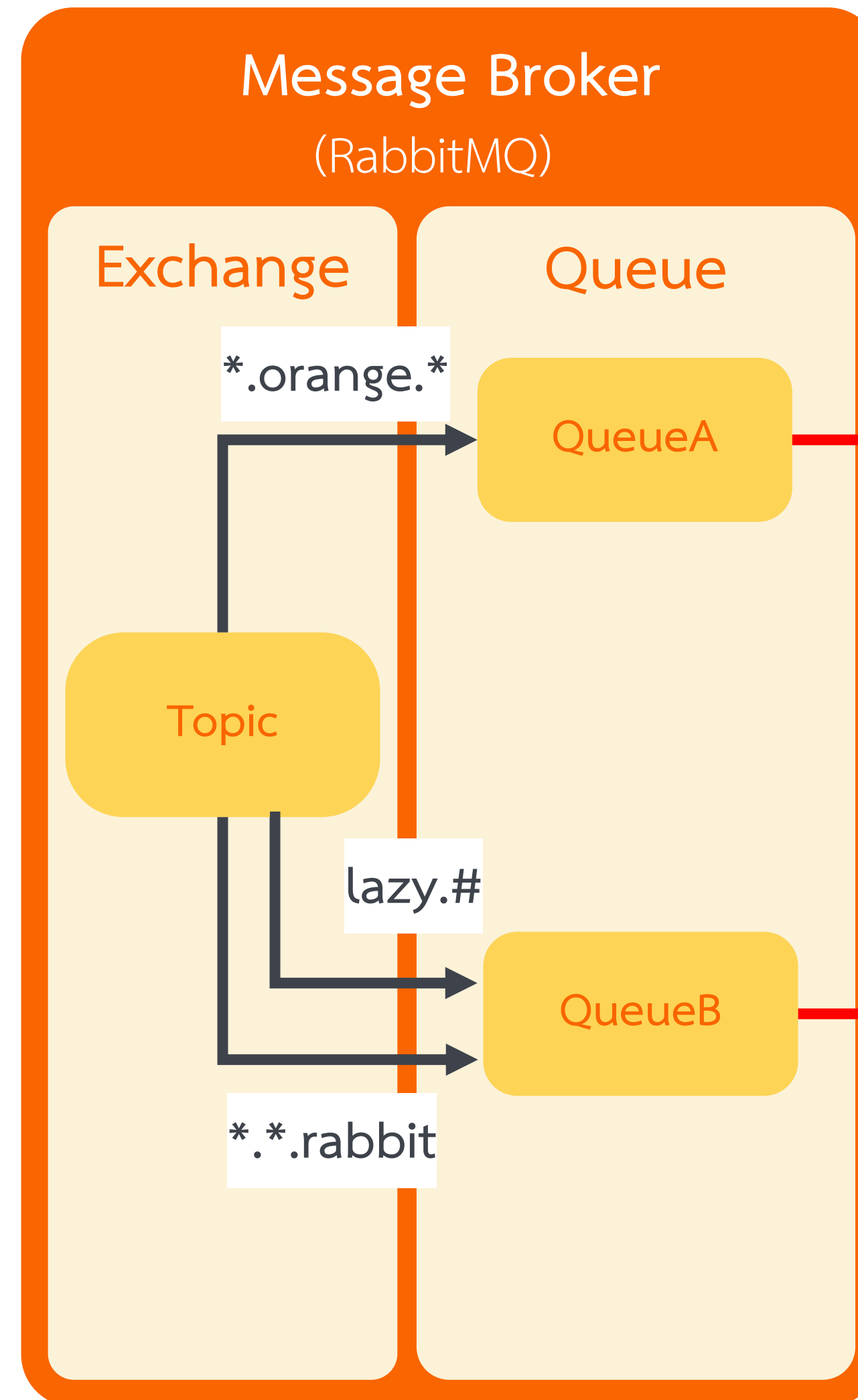


Topic Exchange

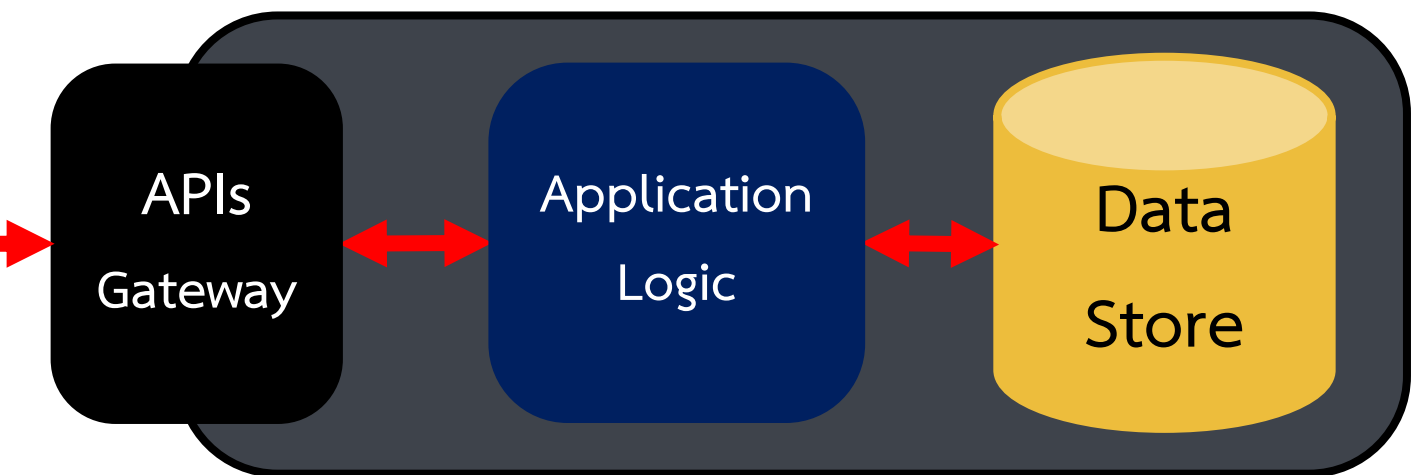
Producers Microservice #1



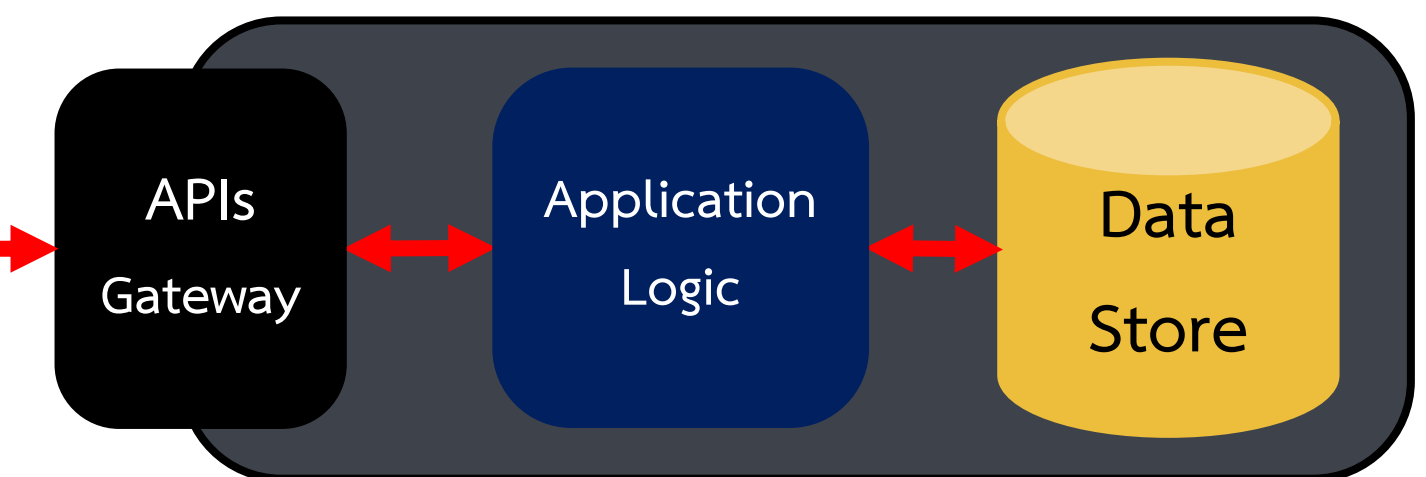
Topic Exchange เป็นการแลกเปลี่ยน Message ระหว่าง Producers กับ Consumers โดยการกำหนด Routing Pattern ในส่วนของ Binding Key เพื่อใช้กำหนดผู้รับปลายทาง และนำ Message มาสร้างสำเนาและส่งไปหาคิวของผู้รับปลายทางเท่านั้น ซึ่งสามารถเรียกว่า “**Multicast** Routing Of Messages”



Consumers Microservice #1

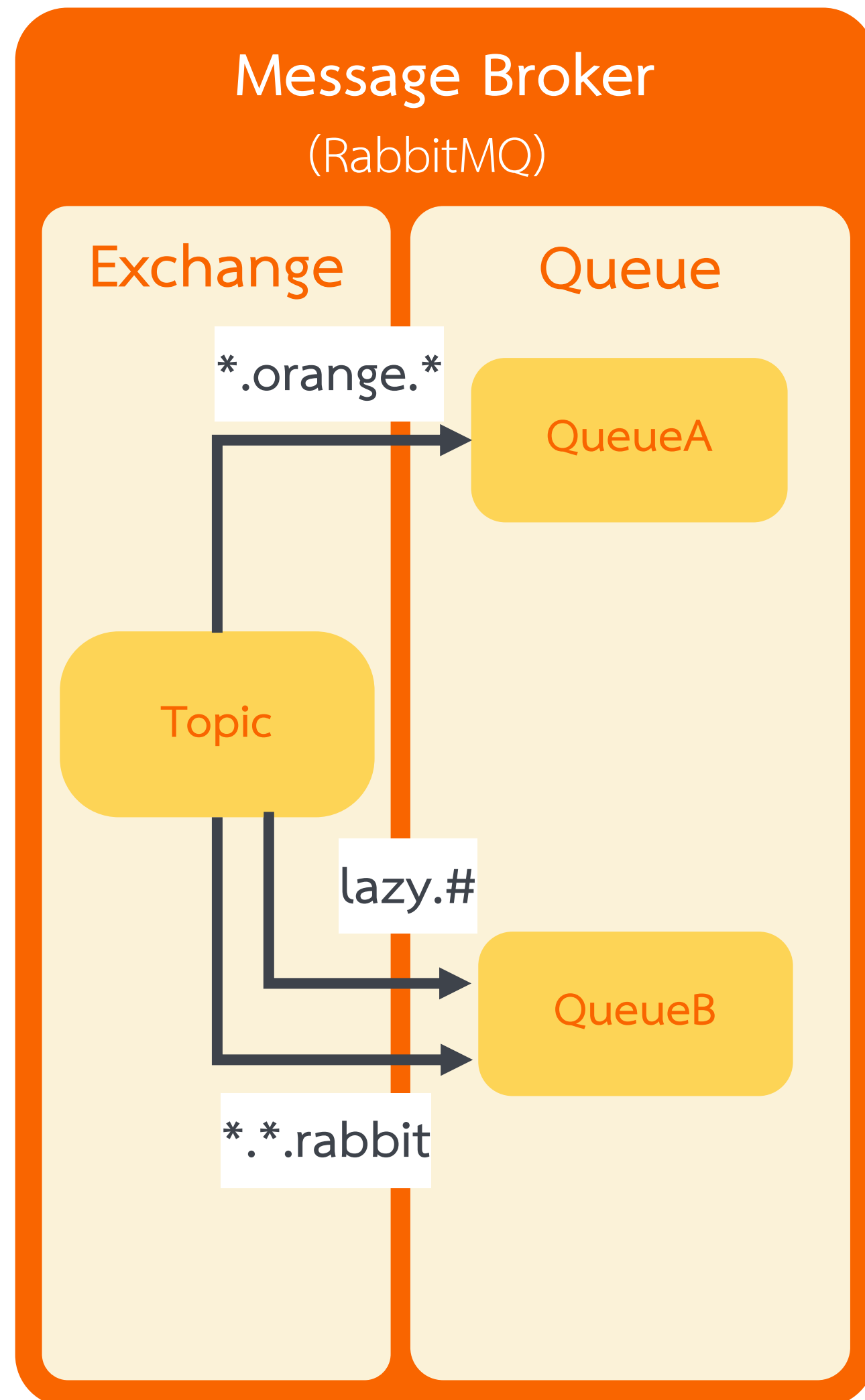


Consumers Microservice #2





หลักการ Routing Pattern



Routing Pattern สามารถสร้างโดยอาศัย 3 เครื่องหมายต่อไปนี้

- `.` ใช้บ่งบอกถึง การแบ่งแยกระหว่าง word
- `*` ใช้บ่งบอกถึง ต้องมีหนึ่ง word (only one)
- `#` ใช้บ่งบอกถึงจะ (i) ไม่มีหรือ (ii) สามารถมี word ตามมากี่อันก็ได้ (zero to many)

ตัวอย่างเช่น

Routing Pattern	Queue	Routing Pattern	Queue
quick.orange.rabbit	A, B	lazy.orange.elephant	A, B
lazy.pink.rabbit	B	quick.orange.fox	A
lazy.brown.fox	B	quick.brown.fox	-
orange	-	quick.orange.male.rabbit	-

หมายเหตุ กรณีที่ Routing Pattern ในส่วนของ binding key เป็น `#` จะกลายเป็นการทำงานแบบ Fanout Exchange นอกจากนี้



กำหนดค่า RabbitMQ แบบ Topic Exchange

ขั้นที่ 1 เลือกเมนู Exchanges

RabbitMQ™

RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-18 14:31:05 Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest Log out

Overview

Connections

Channels

Exchanges

Queues

Admin

Exchanges

▼ All exchanges (7)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Displaying 7 items , page size up to: 100

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

(2)เลือกเพื่อสร้างรูปแบบการ Exchange

HTTP API

Server Docs

Tutorials

Community Support

Community Slack

Commercial Support

Plugins

GitHub

Changelog



กำหนดค่า RabbitMQ แบบ Topic Exchange

ขั้นที่ 2 สร้าง Exchanges

Refreshed 2021-10-18 14:31:05 Refresh every 5 seconds Virtual host All Cluster rabbit@Mac-mini-khxng-Taravichet User guest Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

▼ All exchanges (7)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

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	I D			
amq.topic	topic	D			

► Add a new exchange

▼ Add a new exchange

Name: (3) ระบุชื่อ "Exchanges"

Type: (4) เลือกชนิด "topic"

Durability:

Auto delete: ?

Internal: ?

Arguments: =

Add Alternate exchange ?


Add exchange (5) กดปุ่ม "Add exchange"

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog



กำหนดค่า RabbitMQ แบบ Topic Exchange

ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

 **RabbitMQ**TM RabbitMQ 3.9.7 Erlang 24.1.2

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

Name	Type	Features	Message rate in	Message rate out
(AMQP default)	direct	D		
MyDirectExchange	direct	D		
MyFanoutExchange	fanout	D	0.00/s	0.00/s
MyTopicExchange	topic	D		

-/-



กำหนดค่า RabbitMQ แบบ Topic Exchange

ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

RabbitMQ™ RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-18 14:35:47 Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

▼ All exchanges (8)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Displaying 8 items , page size up to: 100

Name	Type	Features	Message rate
(AMQP default)	direct	D	
MyTopicExchange	topic	D	
amq.direct	direct	D	
amq.fanout	fanout	F	

คลิกที่

Add binding from this exchange

To queue ▼: Computer (1) ระบุชื่อ "Queue"

Routing key: *.computer.# (2) ระบุ "Routing Pattern"

Arguments: = String ▼

Bind (3) กด "Bind" เพื่อ Route



ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

RabbitMQ™

RabbitMQ 3.9.7

Erlang 24.1.2

Refreshed 2021-10-18 14:35:47

Refresh every 5 seconds

Virtual host

All

Cluster **rabbit@Mac-mini-khxng-Taravichet**

User **guest**

Log out

Overview

Connections

Channels

Exchanges

Queues

Admin

Exchanges

▼ All exchanges (8)

Pagination

Page 1 of 1 - Filter:

Name	Type	Features	Message rate
(AMQP default)	direct	<div>D</div>	
MyTopicExchange	topic	<div>D</div>	
amq.direct	direct	<div>D</div>	
amq.fanout	fanout	<div>F</div>	

▼ Bindings

This exchange

↓

To	Routing key	Arguments	
Computer	*.computer.#		<div>Unbind</div>
Mobile	*.mobile.*		<div>Unbind</div>
TV	#.tv		<div>Unbind</div>

Displaying 8 items , page size up to: 100



Spring Boot กับ RabbitMQ แบบ Topic Exchange

การสร้าง **Publisher** ให้ส่ง Message ไปที่ Broker (RabbitMQ) โดยอาศัยคลาส RabbitTemplate

```
package com.example.demo;
import org.springframework.amqp.rabbit.core.RabbitTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
public class Publisher03 {

    @Autowired
    private RabbitTemplate rabbitTemplate;

    @RequestMapping(value = "/queueUp/{name}", method = RequestMethod.GET)
    public String queueUp(@PathVariable("name") String name){
        System.out.println("Publisher In : " + name );
        rabbitTemplate.convertAndSend("MyTopicExchange", "orange.computer.ant.tv", name);
        return "Success";
    }
}
```

Routing Pattern



Spring Boot กับ RabbitMQ แบบ Topic Exchange

การสร้าง **Consumer** ให้ฟัง Request Message จาก Queue โดยอาศัย RabbitListener

```
package com.example.demo;
import org.springframework.amqp.rabbit.annotation.RabbitListener;
import org.springframework.stereotype.Service;
// เหมือนกับ Fanout Exchange
@Service
public class Consumer03 {

    @RabbitListener(queues = "TV")
    public void queueTV(String name){    System.out.println("In TV Queue : "+ name);    }

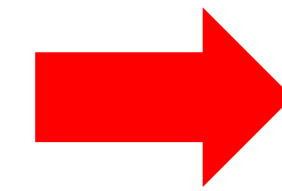
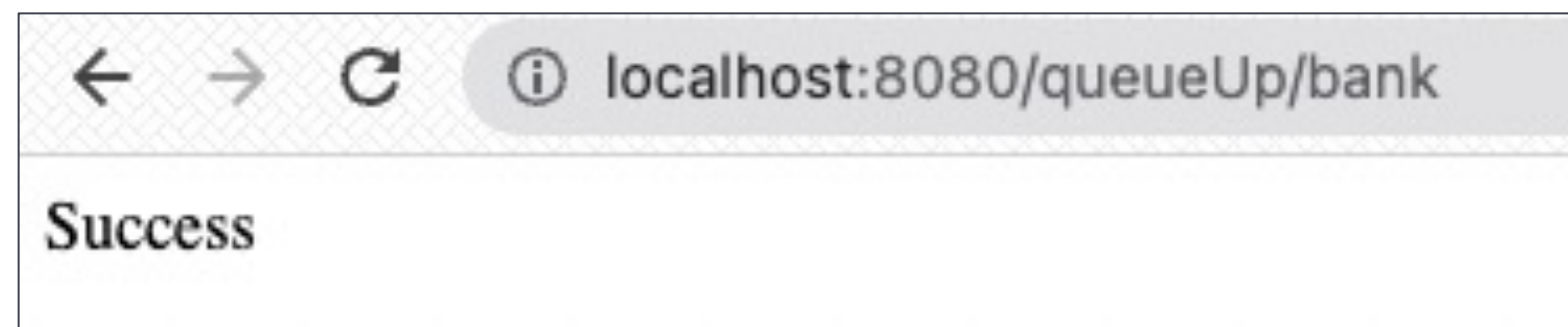
    @RabbitListener(queues = "Computer")
    public void queueCom(String name){    System.out.println("In Computer Queue : "+ name);    }

    @RabbitListener(queues = "Mobile")
    public void queueMobile(String name){    System.out.println("In Mobile Queue : "+ name);    }

}
```



Spring Boot กับ RabbitMQ แบบ Topic Exchange



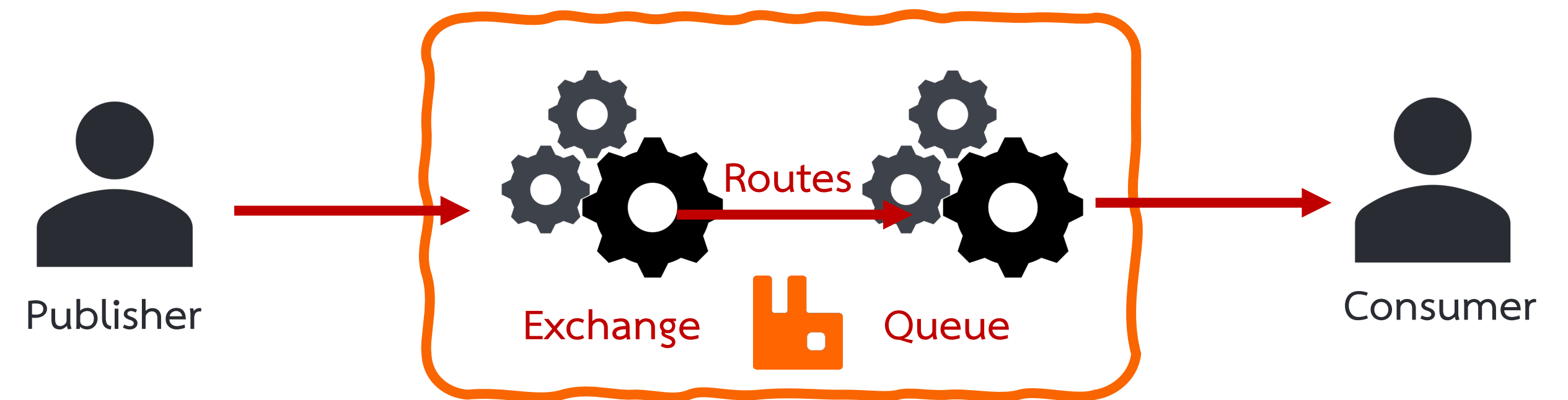
```
Publisher In : bank  
In TV Queue : bank  
In Computer Queue : bank
```

ข้อดีของการส่งแบบ Topic Exchange คือ การ Blind ที่อยู่ของปลายทาง และสามารถสร้าง Pattern ขององค์กรได้



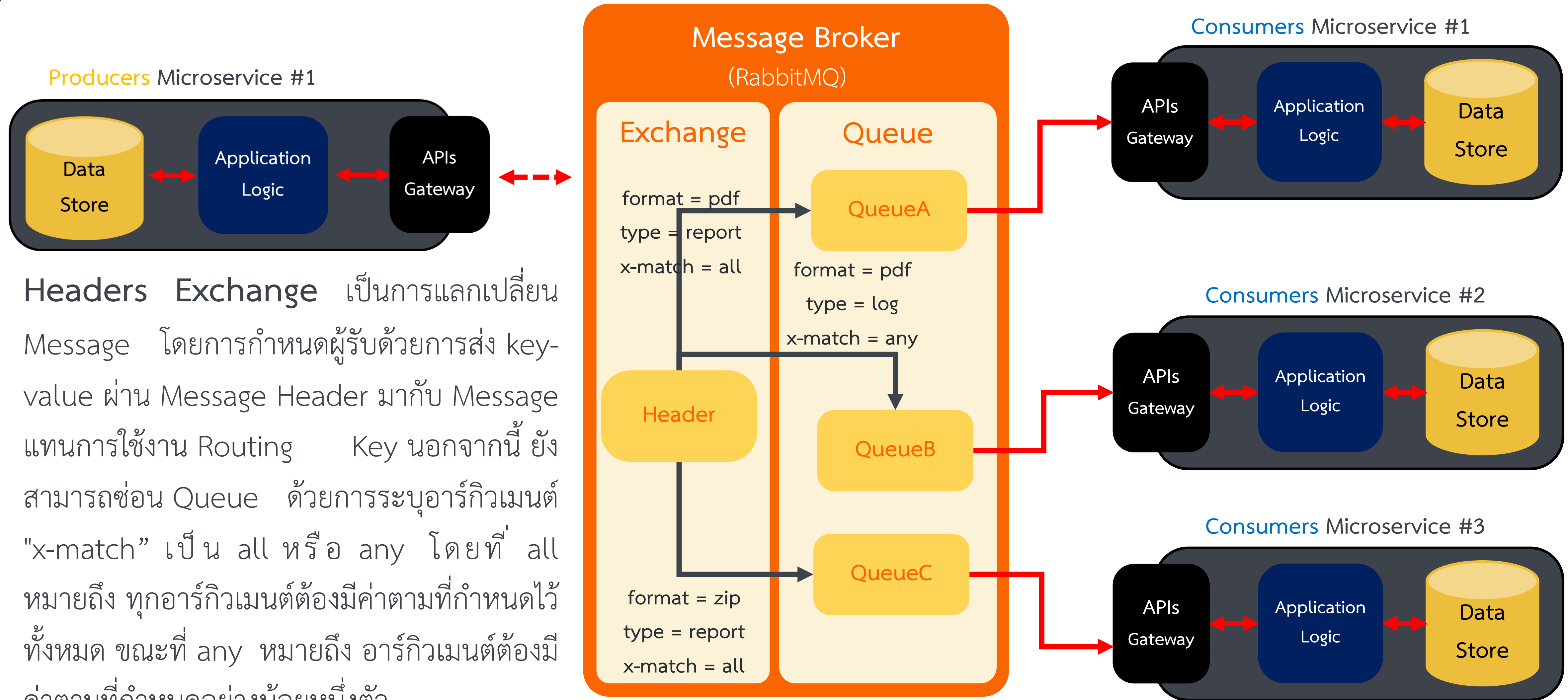
Outline

- Work queues
- Exchange type
 - Direct Exchange (default)
 - Fanout Exchange
 - Topic Exchange
 - Headers Exchange



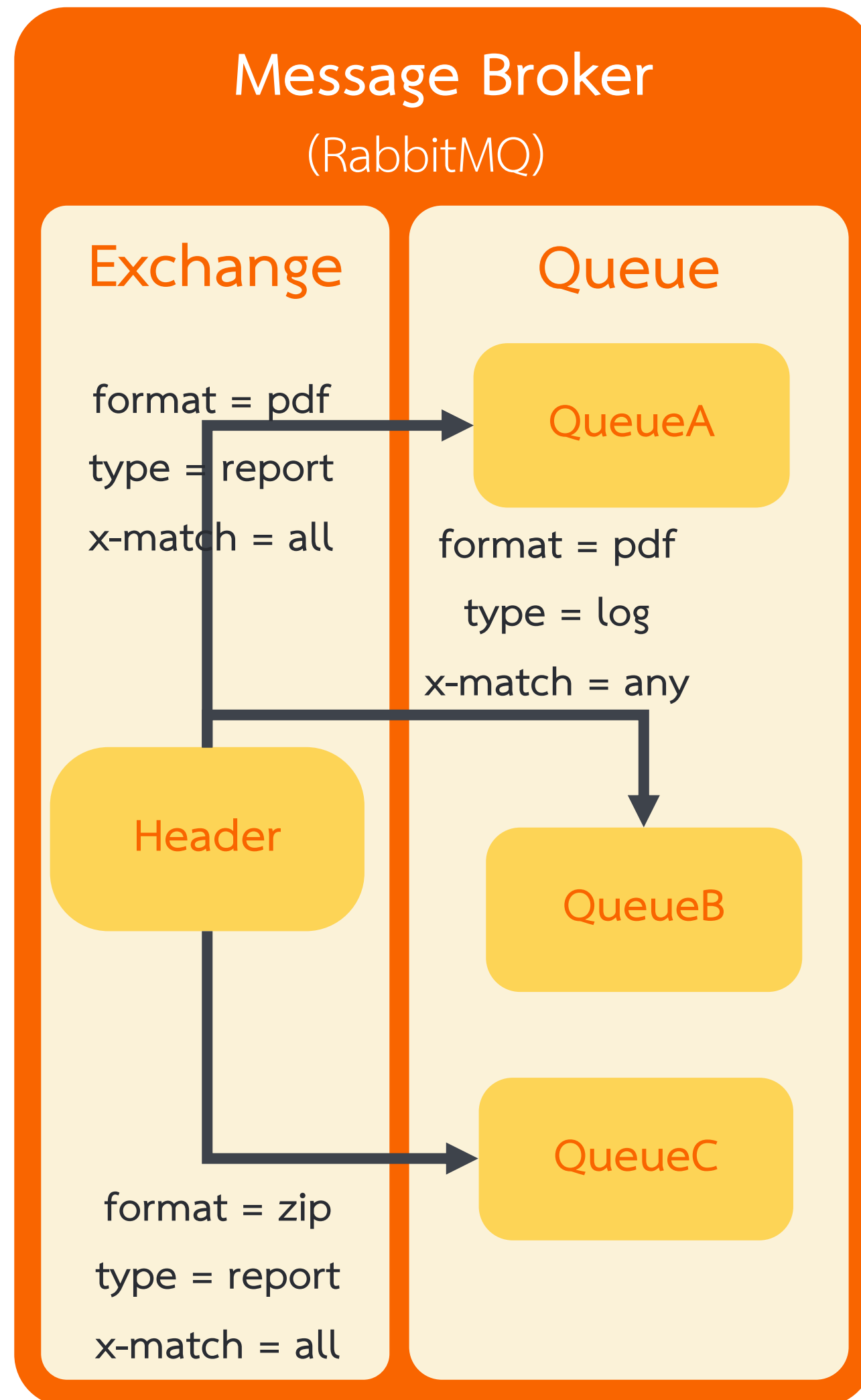


Headers Exchange





หลักการ Routing ของ Headers Exchange



ตัวอย่างเช่น

Header Message	Queue
format = pdf, type = report	A, B
format = pdf	B
format = zip, type = log	B



กำหนดค่า RabbitMQ แบบ Headers Exchange

ขั้นที่ 1 เลือกเมนู Exchanges

RabbitMQ™

RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-18 14:31:05 Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

▼ All exchanges (7)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Displaying 7 items , page size up to: 100

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

(2)เลือกเพื่อสร้างรูปแบบการ Exchange

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog



กำหนดค่า RabbitMQ แบบ Headers Exchange

ขั้นที่ 2 สร้าง Exchanges

RabbitMQ™ RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-18 14:31:05 Refresh every 5 seconds Virtual host All Cluster rabbit@Mac-mini-khxng-Taravichet User guest Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

▼ All exchanges (7)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

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	I			
amq.topic	topic	D			

► Add a new exchange

▼ Add a new exchange

Name: *

Type: ▼

Durability: ▼

Auto delete: ? ▼

Internal: ? ▼

Arguments: = String ▼

Add Alternate exchange ?

Add exchange

(1) คลิกเลือกเมนู “Exchanges”

(2) กดปุ่ม “Add a new exchange”

(3) ระบุชื่อ “Exchanges”

(4) เลือกชนิด “headers”


(5) กดปุ่ม “Add exchange”

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog



กำหนดค่า RabbitMQ แบบ Headers Exchange

ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

 **RabbitMQ**TM RabbitMQ 3.9.7 Erlang 24.1.2

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

► All exchanges (9)

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
MyDirectExchange	direct	D			
MyFanoutExchange	fanout	D	0.00/s	0.00/s	
MyHeadExchange	headers	D			
MyTopicExchange	topic	D	0.00/s	0.00/s	



กำหนดค่า RabbitMQ แบบ Headers Exchange

ขั้นที่ 3 กำหนด Route จาก Exchanges ไปหา Queue

RabbitMQ™ RabbitMQ 3.9.7 Erlang 24.1.2

Refreshed 2021-10-18 14:35:47 Refresh every 5 seconds

Virtual host All

Cluster rabbit@Mac-mini-khxng-Taravichet

User guest Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchanges

▼ All exchanges (8)

Pagination

Page 1 of 1 - Filter:

Name	Type	Features	Message
(AMQP default)	direct	D	
MyHeadExchange	headers	D	คลิกที่นี่
amq.direct	direct	D	
amq.fanout	fanout	F	

Add binding from this exchange

(1) ระบุชื่อ "Queue"

To queue ▼: Computer

Routing key:

(2) ระบุอาร์กิวเมนต์

Arguments:			
canCall	=	false	Boolean ✓
canPlayGame	=	true	Boolean ✓
canWatch	=	true	Boolean ✓
	=		String ✓

(3) กด "Bind" เพื่อ Route

Bind

page size up to: 100



RabbitMQ™

RabbitMQ 3.9.7

Erlang 24.1.2

Refreshed 2021-10-18 14:35:47

Refresh every 5 seconds

Virtual host

All

Cluster **rabbit@Mac-mini-khxng-Taravichet**

User **guest**

Log out

Overview

Connections

Channels

Exchanges

Queues

Admin

Exchanges

▼ All exchanges (8)

Pagination

Page 1 of 1 - Filter:

Name	Type	Features	Message rate
(AMQP default)	direct	<div>D</div>	
MyHeadExchange	headers	<div>D</div>	
amq.direct	direct	<div>D</div>	
amq.fanout	fanout	<div>F</div>	

▼ Bindings

This exchange

⇓

To	Routing key	Arguments	
Computer		<div>canCall: false</div> <div>canPlayGame: true</div> <div>canWatch: true</div> <div>x-match: all</div>	<div>Unbind</div>
Mobile		<div>canCall: true</div> <div>canPlayGame: true</div> <div>canWatch: true</div> <div>x-match: all</div>	<div>Unbind</div>
TV		<div>canCall: false</div> <div>canPlayGame: false</div> <div>canWatch: true</div> <div>x-match: all</div>	<div>Unbind</div>

Displaying 8 items , page size up to: 100



Spring Boot กับ RabbitMQ แบบ Headers Exchange

การสร้าง **Publisher** ให้ส่ง Message ไปที่ Broker (RabbitMQ) โดยอาศัยคลาส RabbitTemplate

```
package com.example.demo;
import org.springframework.amqp.core.*;
import org.springframework.amqp.rabbit.core.RabbitTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.io.*;

@RestController
public class Publisher04 {
    @Autowired
    private RabbitTemplate rabbitTemplate;
    @RequestMapping(value = "/queueUp/{name}", method = RequestMethod.GET)
    public String queueUp(@PathVariable("name") String name){

        // Code อยู่หน้าถัดไป

    }
}
```



Spring Boot กับ RabbitMQ แบบ Headers Exchange

การสร้าง **Publisher** ให้ส่ง Message ไปที่ Broker (RabbitMQ) โดยอาศัยคลาส RabbitTemplate

```
public String queueUp(@PathVariable("name") String name){
    System.out.println("Publisher In : " + name );
    try (ByteArrayOutputStream bos = new ByteArrayOutputStream();
        ObjectOutput out = new ObjectOutputStream(bos);) {
        out.writeObject(name);
        out.flush();
        byte [] byteMessage = bos.toByteArray();
        Message msg = MessageBuilder.withBody(byteMessage) // ต้องการ data type เป็น binary
            .setHeader("canCall",true)
            .setHeader("canPlayGame", true)
            .setHeader("canWatch",true)
            .build();
        rabbitTemplate.send("MyHeadExchange","",msg);
        return "Success";
    } catch (IOException e){
        System.out.println("IO Ex");      return "Fail";
    }
}
```




Spring Boot กับ RabbitMQ แบบ Headers Exchange

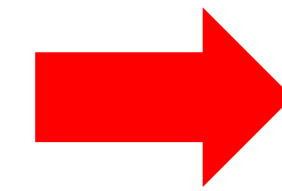
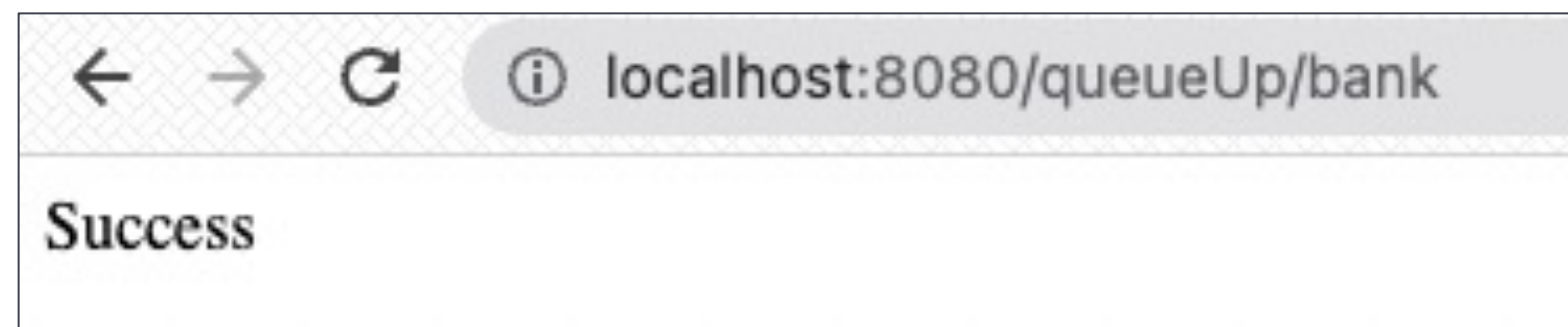
```
package com.example.demo;
import org.springframework.amqp.rabbit.annotation.RabbitListener;
import org.springframework.stereotype.Service;
import java.io.*;

@Service
public class Consumer04 {
    @RabbitListener(queues = "TV")
    public void queueTV(byte[] msg) {
        try (ByteArrayInputStream bis = new ByteArrayInputStream(msg);
             ObjectInput in = new ObjectInputStream(bis);) {
            String name = (String) in.readObject();
            System.out.println("In TV Queue : " + name);
        } catch (ClassNotFoundException | IOException e) {
            System.out.println("IO Ex");
        }
    }

    @RabbitListener(queues = "Computer")
    public void queueCom(byte[] msg) {...}
    @RabbitListener(queues = "Mobile")
    public void queueMobile(byte[] msg) {...}
}
```



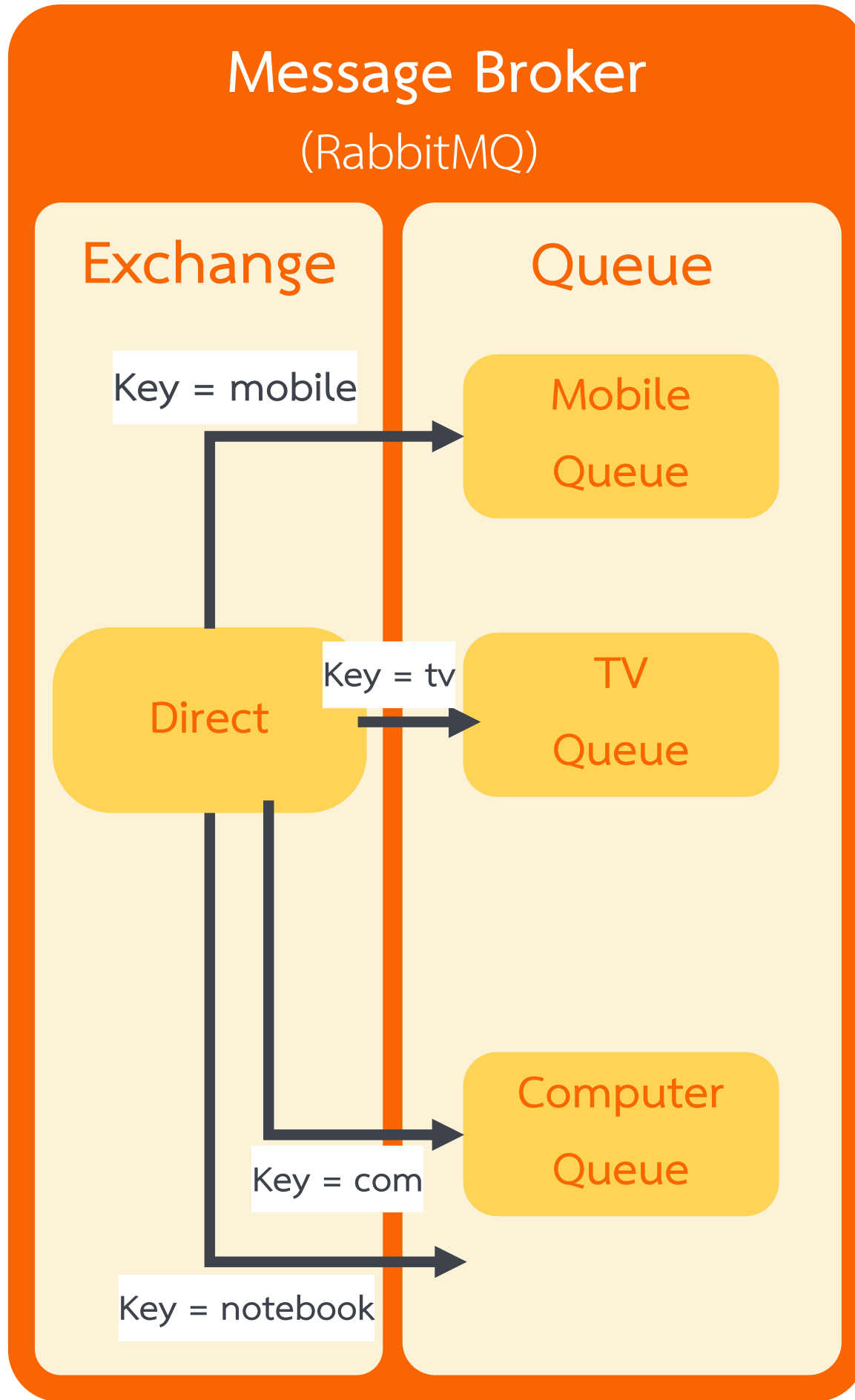
Spring Boot กับ RabbitMQ แบบ Headers Exchange



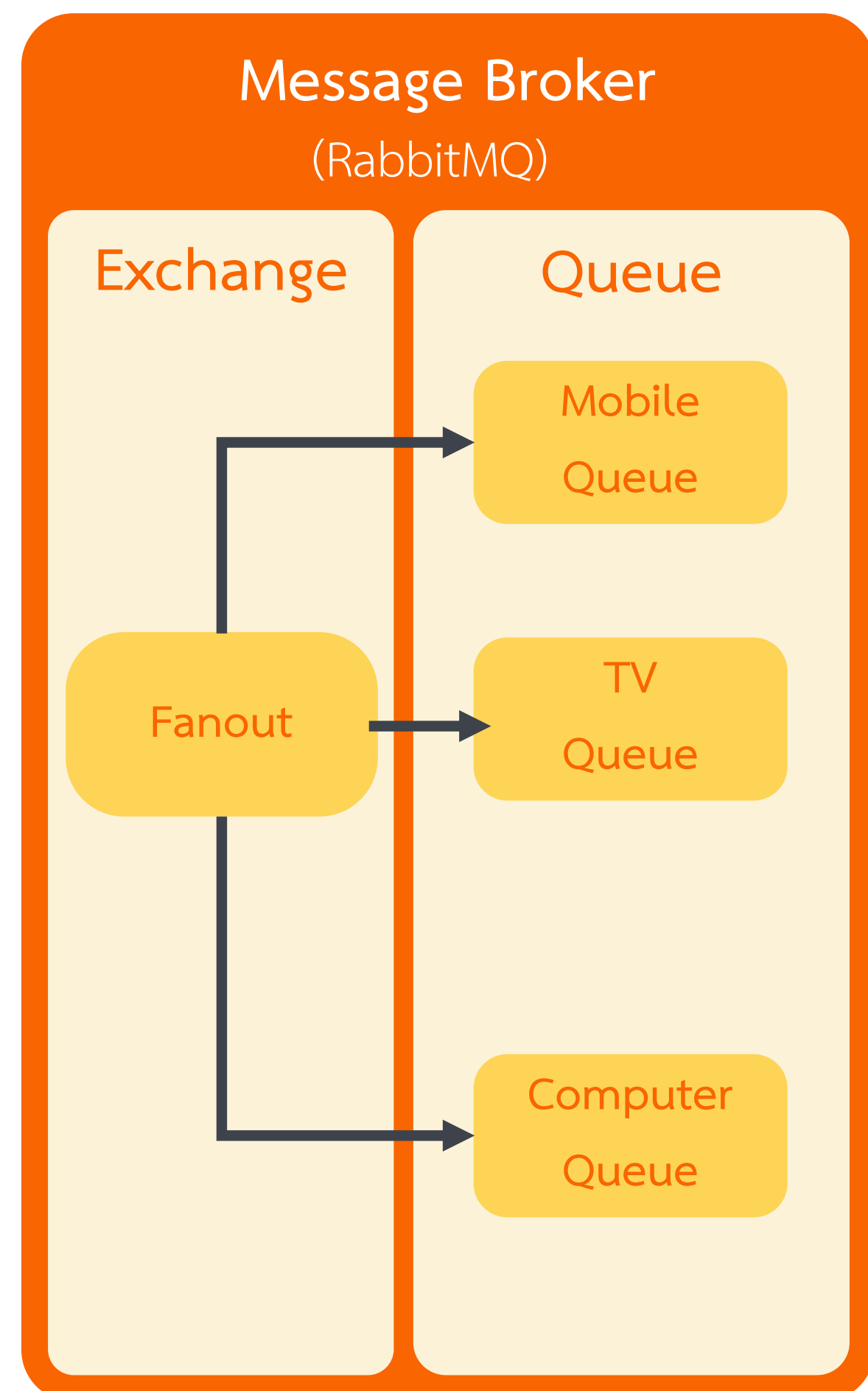
```
Publisher In : bank  
In Mobile Queue : bank
```



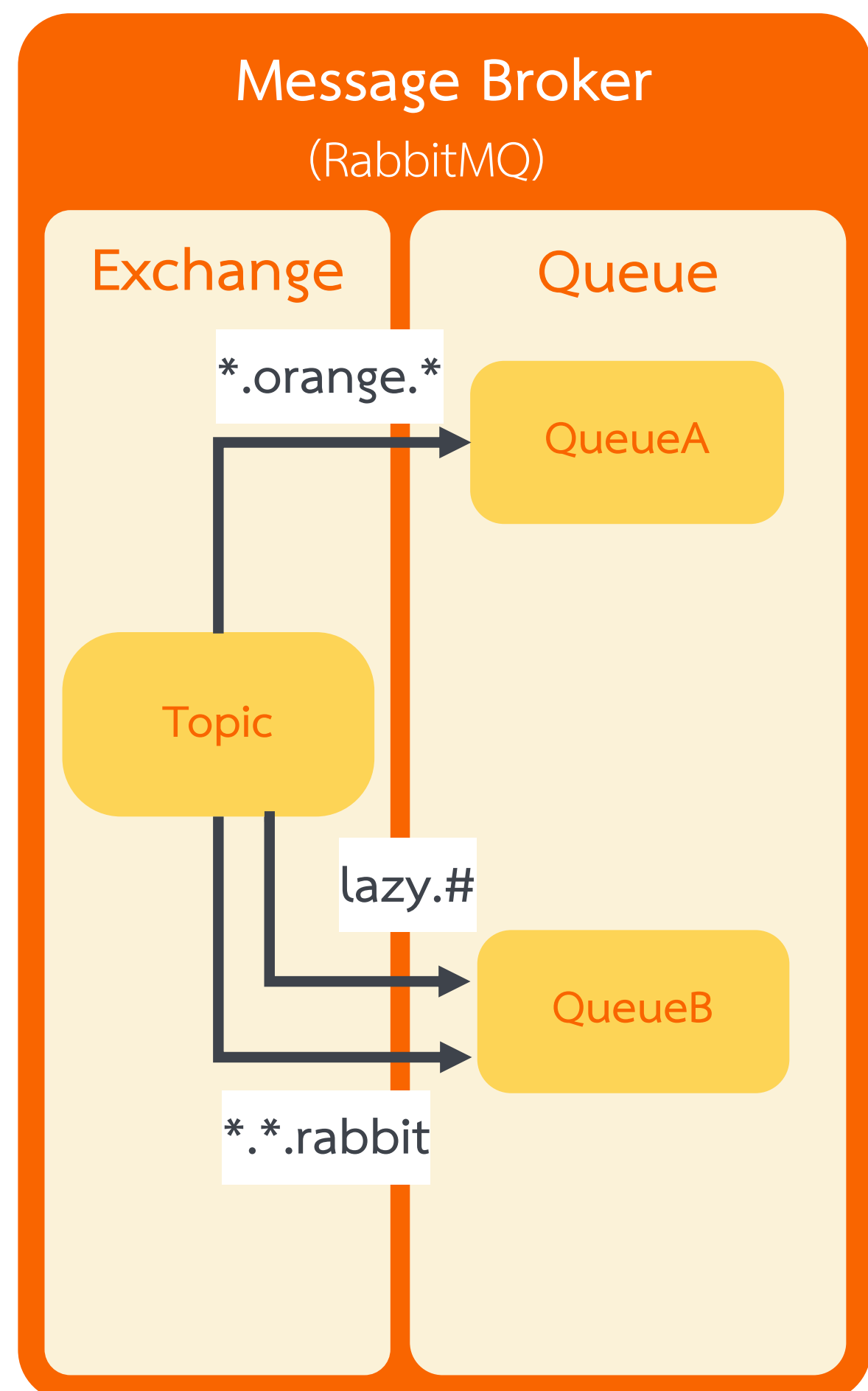
Exchange Type



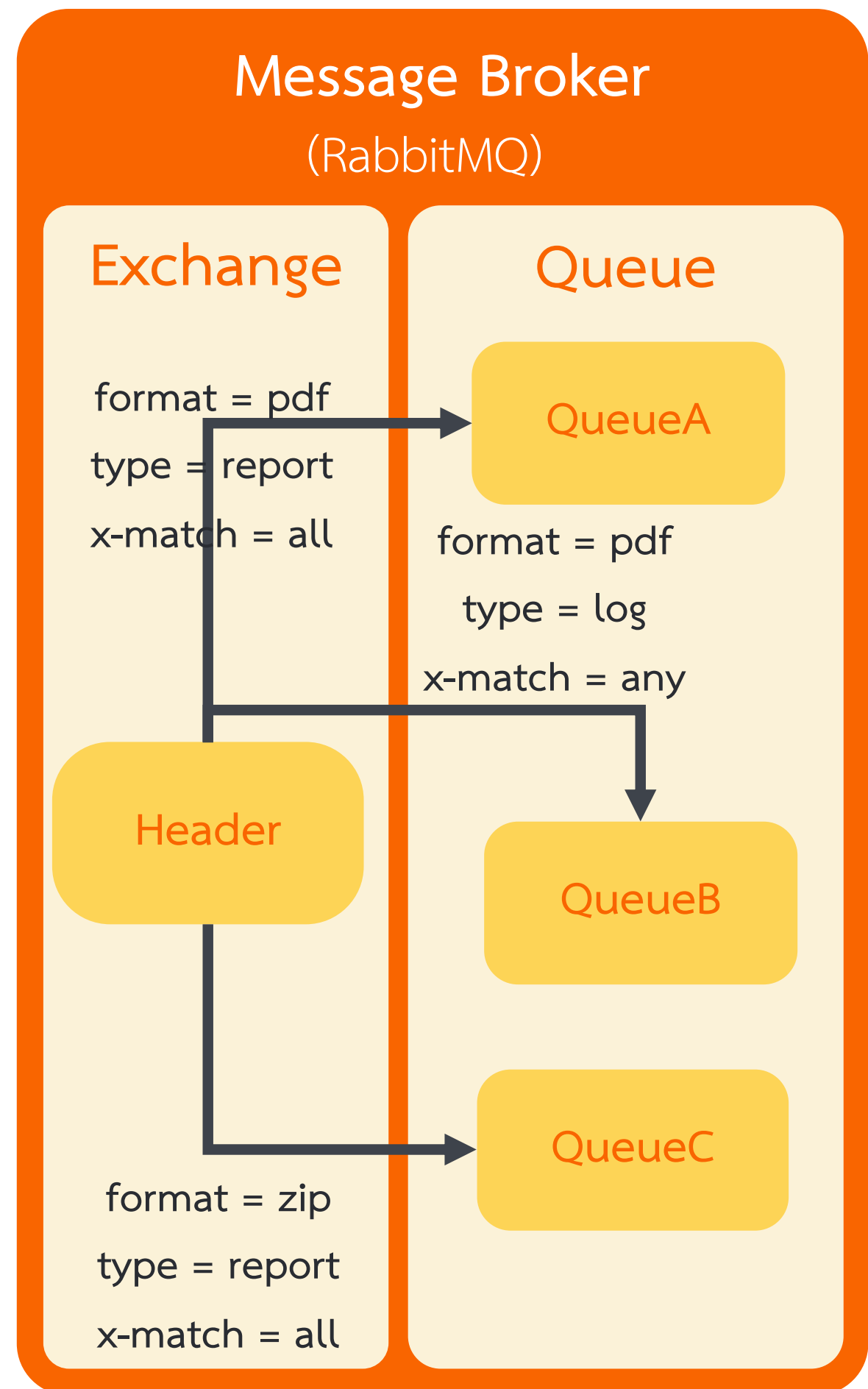
Direct Exchange



Fanout Exchange



Topic Exchange

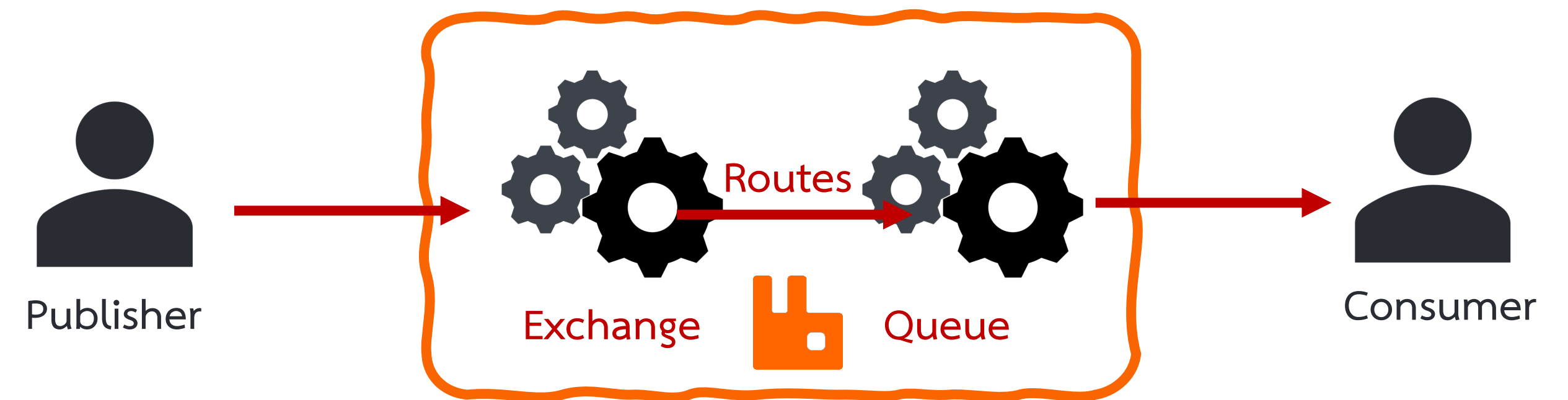


Headers Exchange



Outline

- Work queues
- Exchange type
 - Direct Exchange (default)
 - Fanout Exchange
 - Topic Exchange
 - Headers Exchange



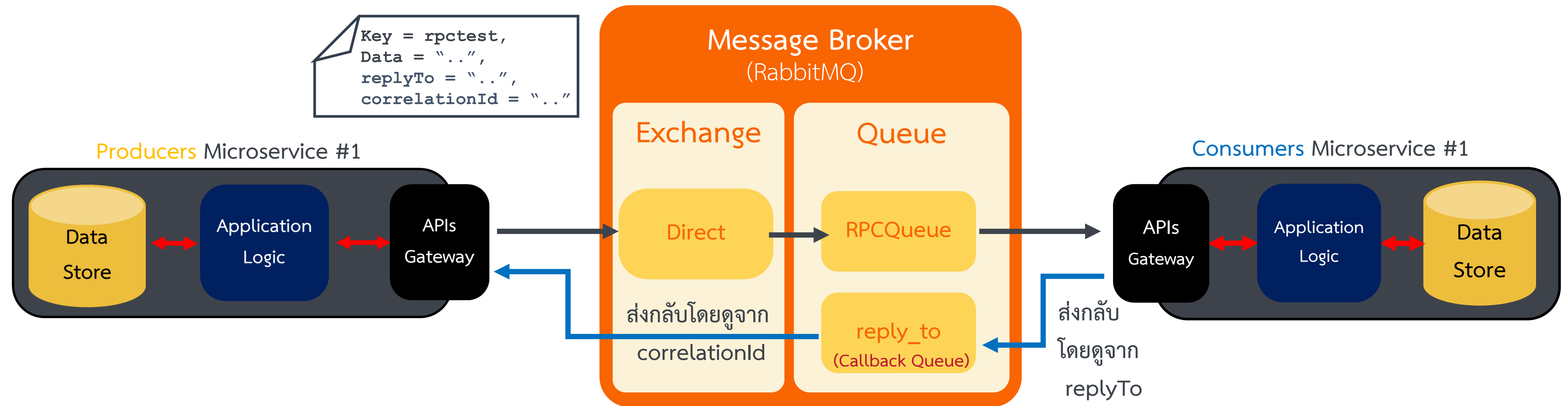


Outline

- การส่งข้อมูลไปกลับ (Request - Reply Pattern) ด้วย RabbitMQ



Request - Reply Pattern



การทำ Remote Procedure Call (RPC) ผ่าน RabbitMQ นั้นเป็นเรื่องง่าย โดยที่ Client ส่ง Request Message ไปพร้อมกับการระบุ **Callback Queue** และ **CorrelationId** จากนั้น Server จะตอบกลับ Response Message มาตามที่อยู่ของ Callback Queue จากนั้น จะส่งกลับไหา Client ซึ่ง RabbitTemplate ของ Spring AMQP เป็นตัวจัดการ และนักศึกษาอาศัยคำสั่ง **convertSendAndReceive()** สำหรับการสื่อสารแบบ RPC



Request - Reply Pattern

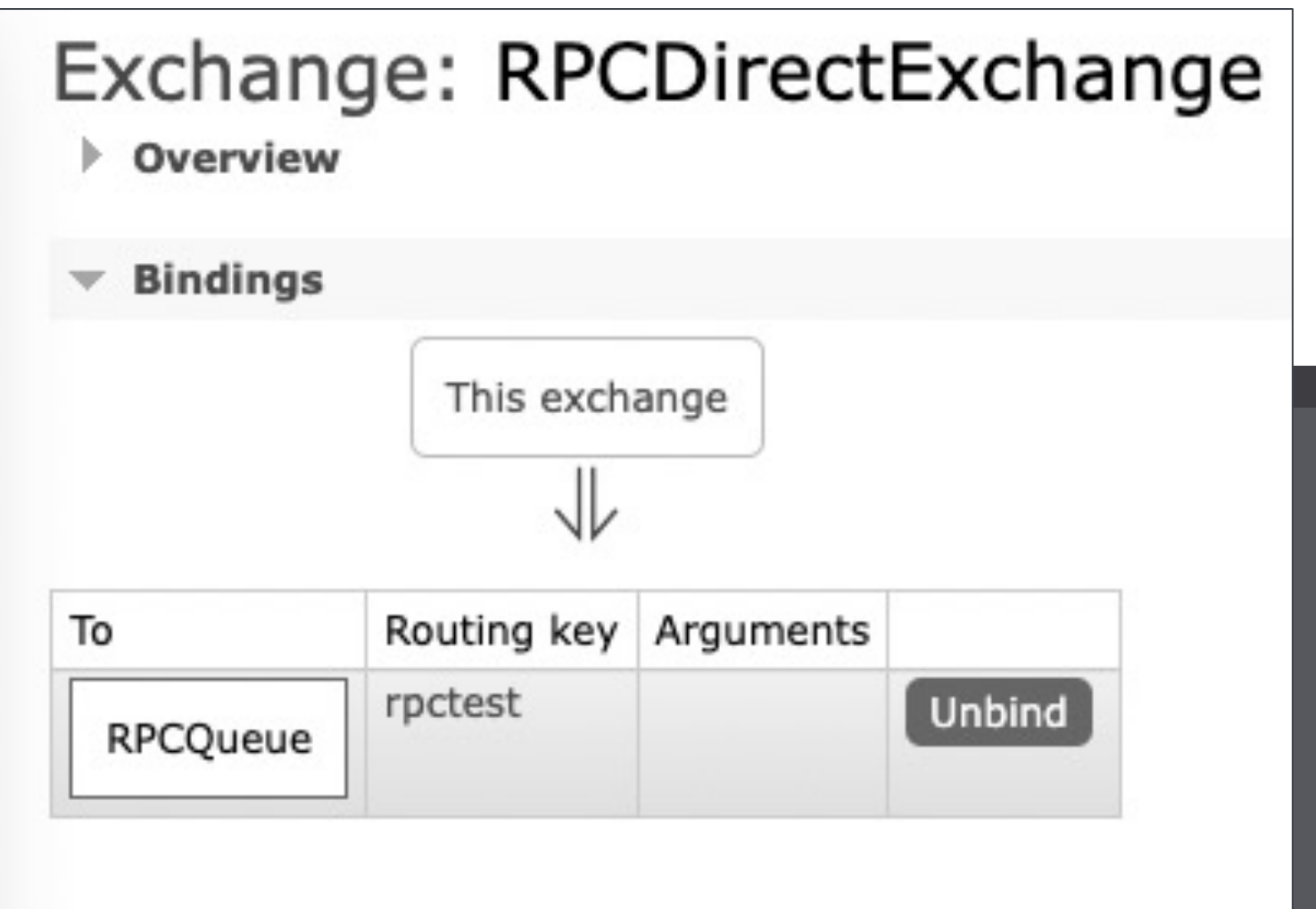
ตัวอย่างโปรแกรมฝั่ง Server

```
@RestController
public class TestRPC {
    @Autowired
    private RabbitTemplate rabbitTemplate;

    @RequestMapping(value = "/hi/{name}", method = RequestMethod.GET)
    public String getHi(@PathVariable("name") String name) {

        Object str = rabbitTemplate
            .convertSendAndReceive("RPCDirectExchange", "rpctest", name);

        return ((String) str);
    }
}
```



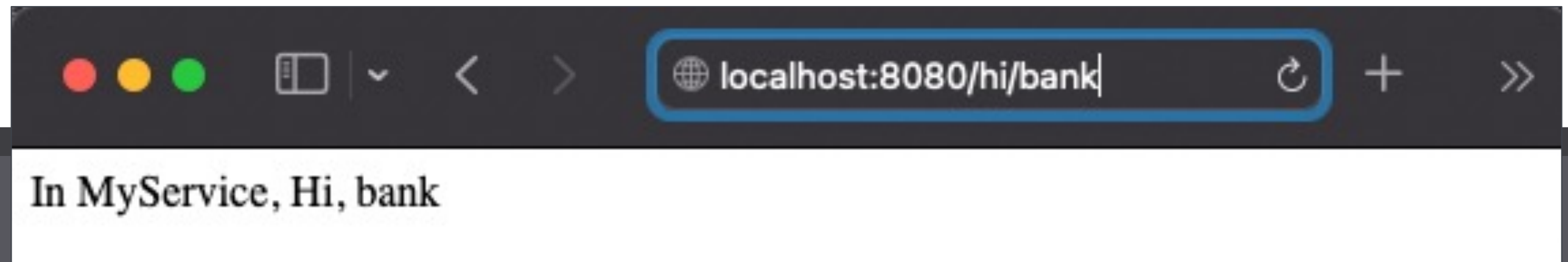


Request - Reply Pattern

ตัวอย่างโปรแกรมฝั่ง Client

```
@Service
public class MyService {

    @RabbitListener(queues = "RPCQueue")
    public String hiMyName(String name) {
        return "In MyService, Hi, "+ name;
    }
}
```



ถ้านักศึกษาสังเกตจะพบ การสื่อสารแบบ RPC มีขั้นตอนไม่แตกต่างจากการสื่อสารแบบ Direct, Fanout, Header และ Topic Exchange ทั้งในส่วนของโปรแกรมและการกำหนดค่าใน RabbitMQ แต่อย่างไรก็ตาม การสื่อสารแบบ RPC รองรับเพียง Direct Exchange เท่านั้น