**rabbitmq五种工作模式和交换机类型**

**1.交换机类型**

rabbitmq应对不同的消息生产和消息消费的场景,使用不同的交换机类型:direct(routing路由模式),fanout(发布订阅),主题模式(topic,特殊的路由)

**2.rabbitmq五种场景**

**2.1简单模式(一对一)**

图示

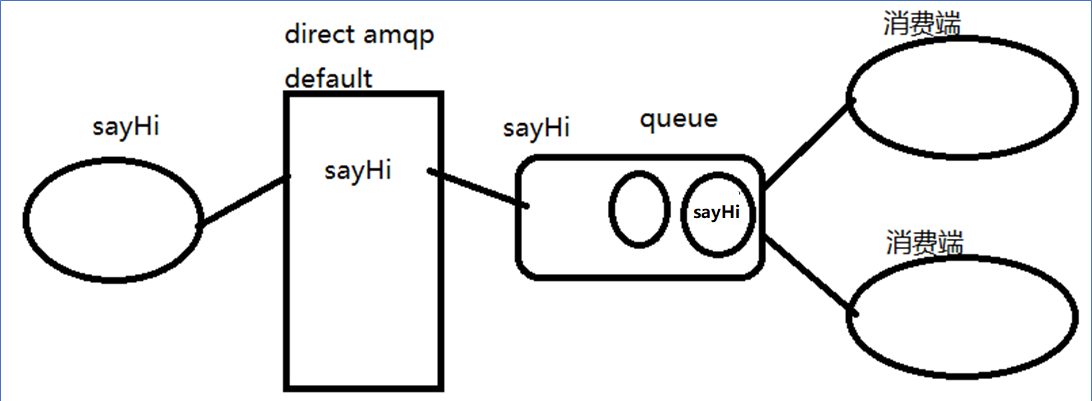
描述已自动生成

在一次消息的生产消费过程中,一个生产者对应一个消费者,形成一发一接的结构;

这里使用的是路由交换机,路由交换机需要绑定队列的名称作为路由Key(路由交换机可以使用默认的AMQP default交换机)

应用场景:消息软件,短信/微信.

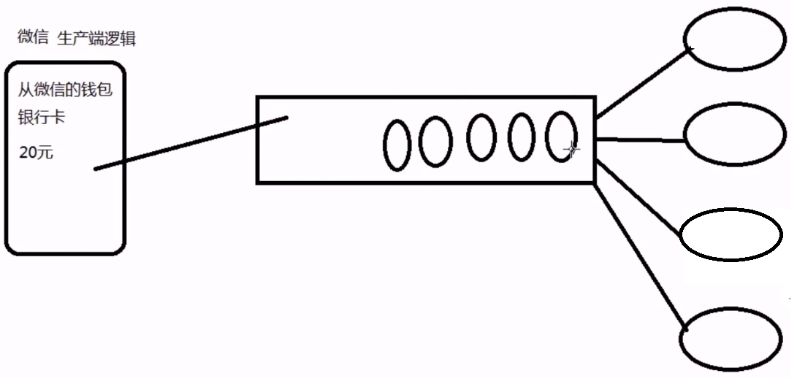
**2.2工作模式(一发多接,资源争抢)**

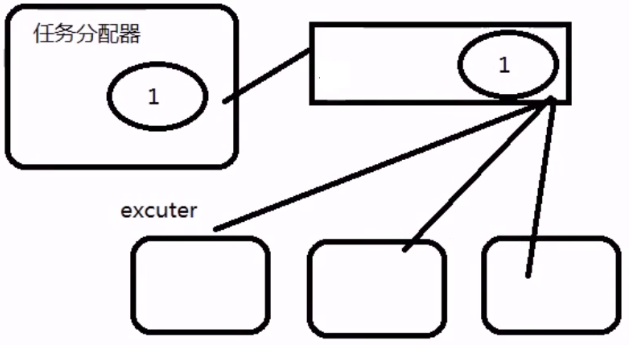


在生产消息发送到队列,之后由多个消费端同时争抢.

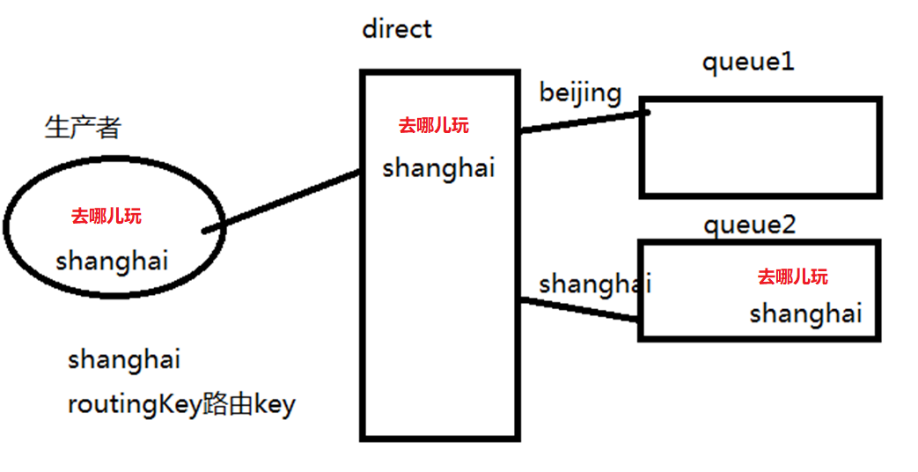
谁的空闲率最大,谁争抢到的可能性就大;交换机可以是任意类型的(代码测试阶段还是AMQP default);

应用场景:抢红包.





**2.3路由模式(direct)**



路由模式不关心后端消费者数量,如果是一个消费者监听一个队列就是简单模式,如果是多个消费者监听一个队列就是争抢模式;

使用路由模式的交换机,可以自定义不同的**路由key绑定**不同队列,在消息中携带一个路由key,通过交换机后匹配后端的队列;

场景: error错误,通过筛选打印在开发人员的屏幕上;

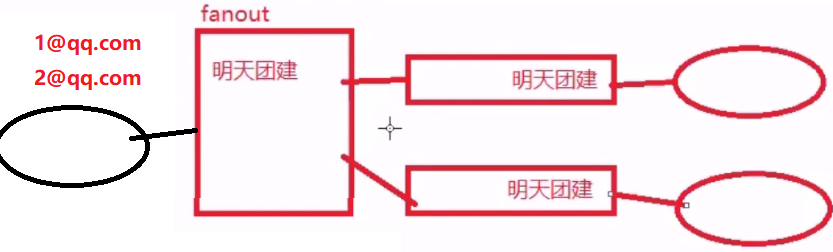
**2.4发布订阅**

图示

描述已自动生成

一条消息不携带具体路由key,经过fanout模式的交换机发送给后端绑定的所有队列;

场景:邮件群发,广告(qq)



**2.5主题模式**

图示

描述已自动生成

主题是一种特殊的路由模式,体现在后端队列绑定topic类型交换机时,可以不使用具体路由key,可以使用匹配符号

\*:表示一个字符串

#:表示任意长度任意字符串;

easymall.\*:可以匹配到消息路由: easymall.product,easymall.user,

easymall.user.save(不能匹配到的)

easymall.#:可以匹配到消息路由:

easymall.product,easymall.dasfas.daskf.sdfklsad.sdf(只要以easymall.开始都能匹配上);

hardmall.kaka(匹配不上了)

场景:多级消息发送接收的路由过程可以使用topic

**快递**

图标

描述已自动生成

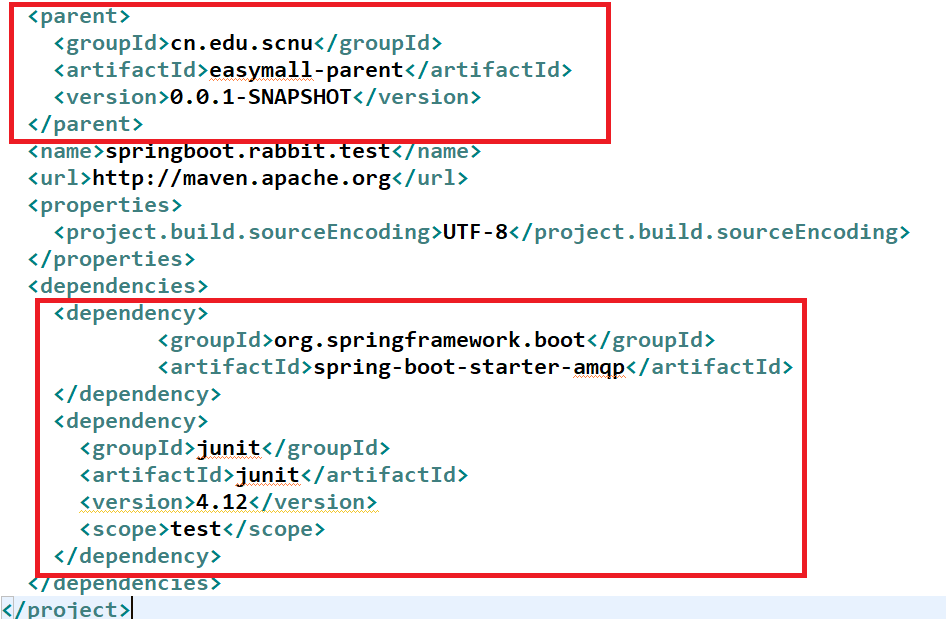
**3.五种工作模式在代码实现**

## 3.1.简单模式代码

### 3.1.1 创建quickstart工程

springboot.rabbit.test

### 3.1.2 继承父级工程，依赖springboot整合依赖



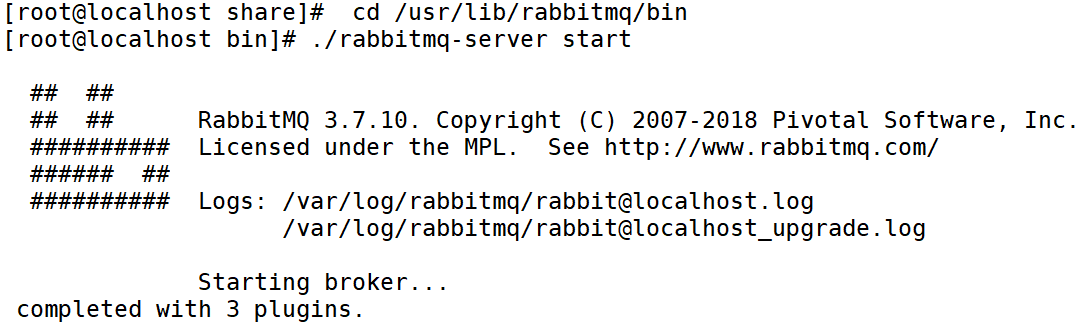
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-amqp</artifactId>

</dependency>

### 3.1.2 启动rabbitMQ



### 3.1.3 实现简单模式代码

**1.初始化信道对象**

图形用户界面, 文本, 应用程序

描述已自动生成

**2.消息发送和接收**

**消息发送端**

图形用户界面, 文本

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

文本

描述已自动生成

## 3.2 .工作模式(争抢资源)

一个生产者发送消息,多个消费者同时监听,形成争抢结构;

### 3.2.1发送端

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 聊天或短信

描述已自动生成

**api介绍:**

**queueDeclare参数意义:**

queue:String类型,代表当前声明的queue名称

durable:Boolean类型,表示当前队列创建之后,声明之后是否持久化(rabbitmq宕机之后是否重启恢复),true表示持久化宕机之后重启可以恢复队列,false表示不作持久化

exclusive:Boolean类型,表示队列是否专属.true表示专属,只有当前连接对象创建的信道能够操作队列,其他连接对象不能,false表示不专属谁都能用。

autoDelete:Boolean,是否自动删除,最后一个channel使用完毕队列,channel关闭之后,是否删除,true自动删,false表示不删

arguments:Map类型,表示queue其他属性设置,例如队列可存储的消息长度,消息存储在队列是否有超时时间

**basicPublish方法:发布消息**

exchange:交换机名称

routingKey:消息发布时携带的路由key,交换机会根据路由key决定消息的目的队列

props:BasicProperties类型,可以通过一个对象定义消息的各种属性,例如,可以指定消息也持久化(必须配合队列)

body:消息体,byge[]数组

### 3.2.2消费端

文本

描述已自动生成

**api方法**

basicConsume:绑定消费者对象与队列,定义回执的逻辑;

channel.basicConsume(queue, autoAck, callback)

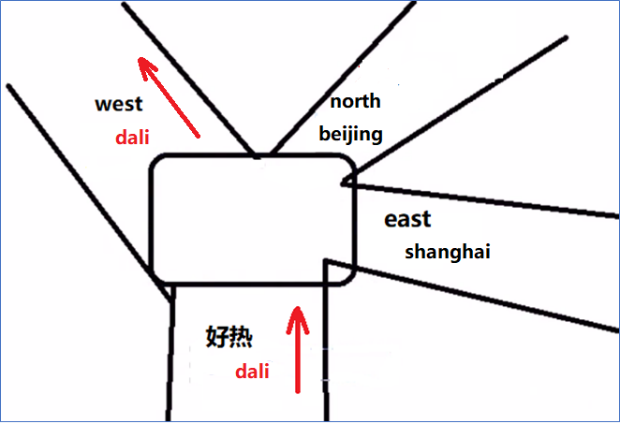
queue:消费对象绑定的队列

callback:回执使用的消费者对象

autoAck:Boolean类型,是否自动回执确认(ack)

true自动确认,一拿到消息就确认,false不自动确认,手动确认

## 3.3 路由模式direct

****

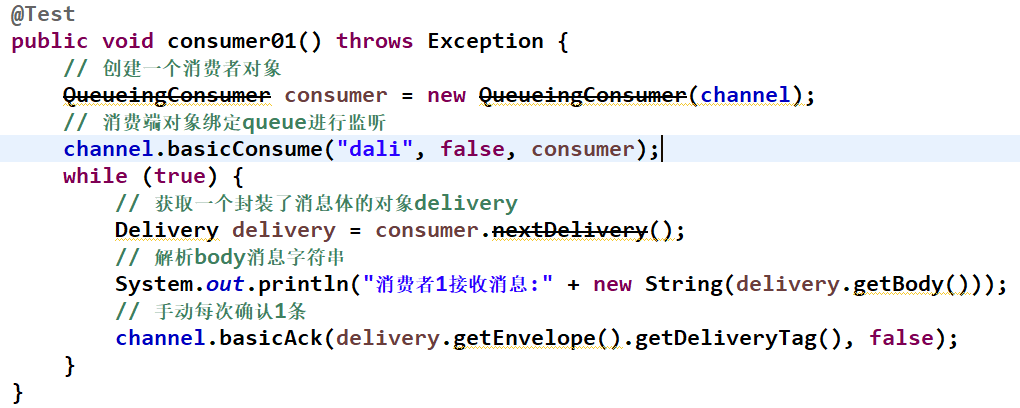
自定义一个路由交换机。

生产者：

文本, 信件

描述已自动生成

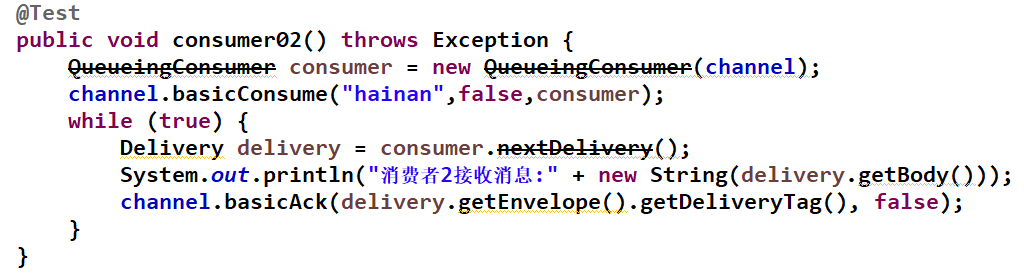
消费者1



图片包含 文本

描述已自动生成

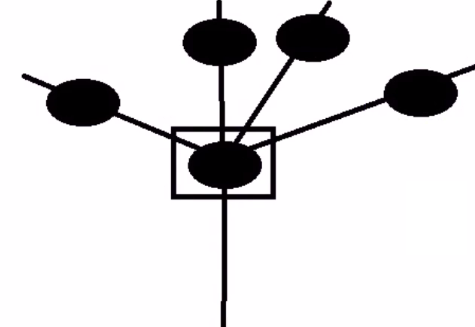
消费者2



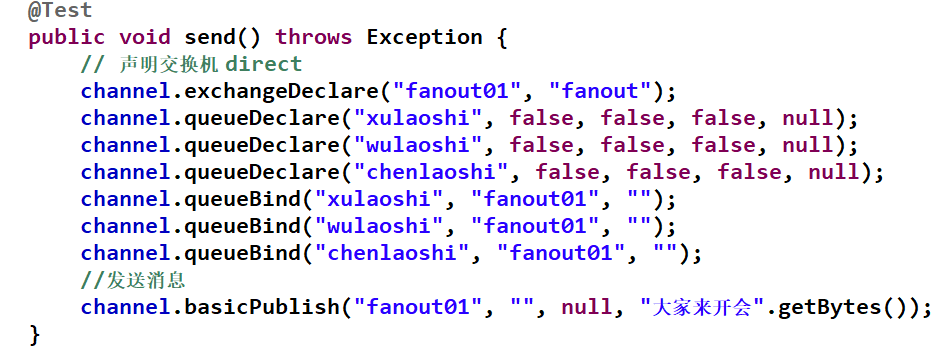
图片包含 文本

描述已自动生成

## 3.4.发布订阅

 图示

描述已自动生成



## 3.5.主题模式

主题模式是路由的一种特殊情况.

后端的队列绑定主题模式的交换机不再需要使用具体的routingKey，可以使用匹配的符号,\*一个字符串,#表示任意长度任意个字符串

