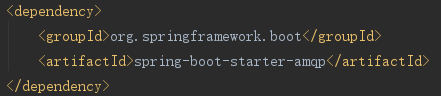
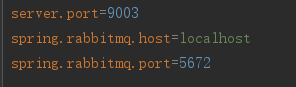
[超详细的RabbitMQ入门，看这篇就够了！-阿里云开发者社区 (aliyun.com)](https://developer.aliyun.com/article/769883)

**4、Direct发布模式**

**核心依赖及application.properties（除端口外不变）**



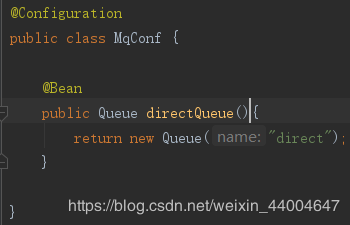


**生产者工程**

4.1、创建消息队列

两种方式：

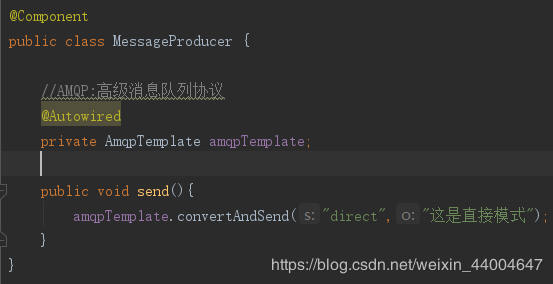
（1）@Configuration和@Bean配置队列conf，指定



4.2、发送端

API：amqpTemplate.convertAndSend("队列名"，“消息内容”)

此处队列名必须与创建的队列一致。



4.3、接收端

@RabbitListener(queues = "direct")：监听器监听指定队列



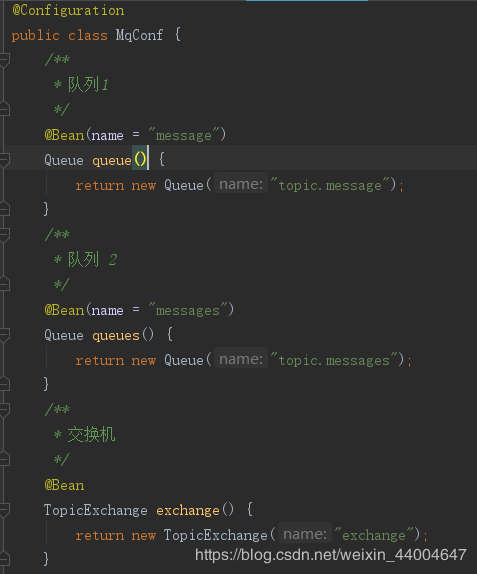
**5、Topic发布模式**

**生产者工程**

5.1、创建消息队列

API：BindingBuilder.bind(指定队列).to(交换机).with(路由键);

路由键相当于队列名

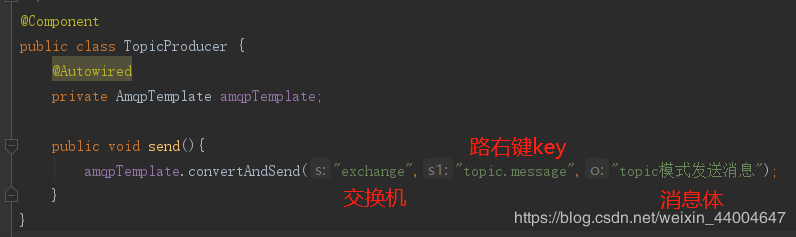




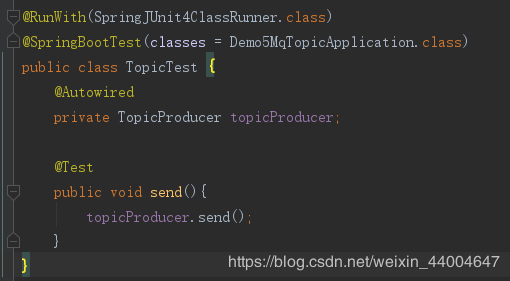
5.2、发送端：

API：amqpTemplate.convertAndSend("交换机名"，“路由键”，“消息内容”)

RabbitMQ将会根据第二个参数去寻找有没有匹配此规则的队列,如果有,则把消息给它,如果有不止一个,则把消息分发给匹配的队列(每个队列都有消息!)



5.3、测试类



6、Fanout发布模式

广播：发送到路由器的消息会使得绑定到该路由器的每一个Queue接收到消息,这个时候就算指定了Key,或者规则(即上文中convertAndSend方法的参数2),也会被忽略!

交换机类型：FanoutExchange

API：amqpTemplate.convertAndSend("交换机名"，“ ”，“消息内容”)；//路由键被忽略

消费端：只要是绑定到该交换机上的都能收到消息。

============================================================================

详细版本：



----------------------------------------------------------------------------

创建队列、交换机、并把他们俩绑定（设定匹配键）

@Configuration
public class DirectRabbitConfig {
@Bean
public Queue rabbitmqDemoDirectQueue() {
*/\*\*
\* 1、name: 队列名称
\* 2、durable: 是否持久化
\* 3、exclusive: 是否独享、排外的。如果设置为true，定义为排他队列。则只有创建者可以使用此队列。也就是private私有的。
\* 4、autoDelete: 是否自动删除。也就是临时队列。当最后一个消费者断开连接后，会自动删除。
\* \*/*
return new Queue(RabbitMQConfig.RABBITMQ\_DEMO\_TOPIC, true, false, false);
}
@Bean
public DirectExchange rabbitmqDemoDirectExchange() {
*//Direct交换机*
return new DirectExchange(RabbitMQConfig.RABBITMQ\_DEMO\_DIRECT\_EXCHANGE, true, false);
}
@Bean
public Binding bindDirect() {
*//链式写法，绑定交换机和队列，并设置匹配键*
return BindingBuilder
*//绑定队列*
.bind(rabbitmqDemoDirectQueue())
*//到交换机*
.to(rabbitmqDemoDirectExchange())
*//并设置匹配键*
.with(RabbitMQConfig.RABBITMQ\_DEMO\_DIRECT\_ROUTING);
}
}

------------------------------------------------------------------------------------------