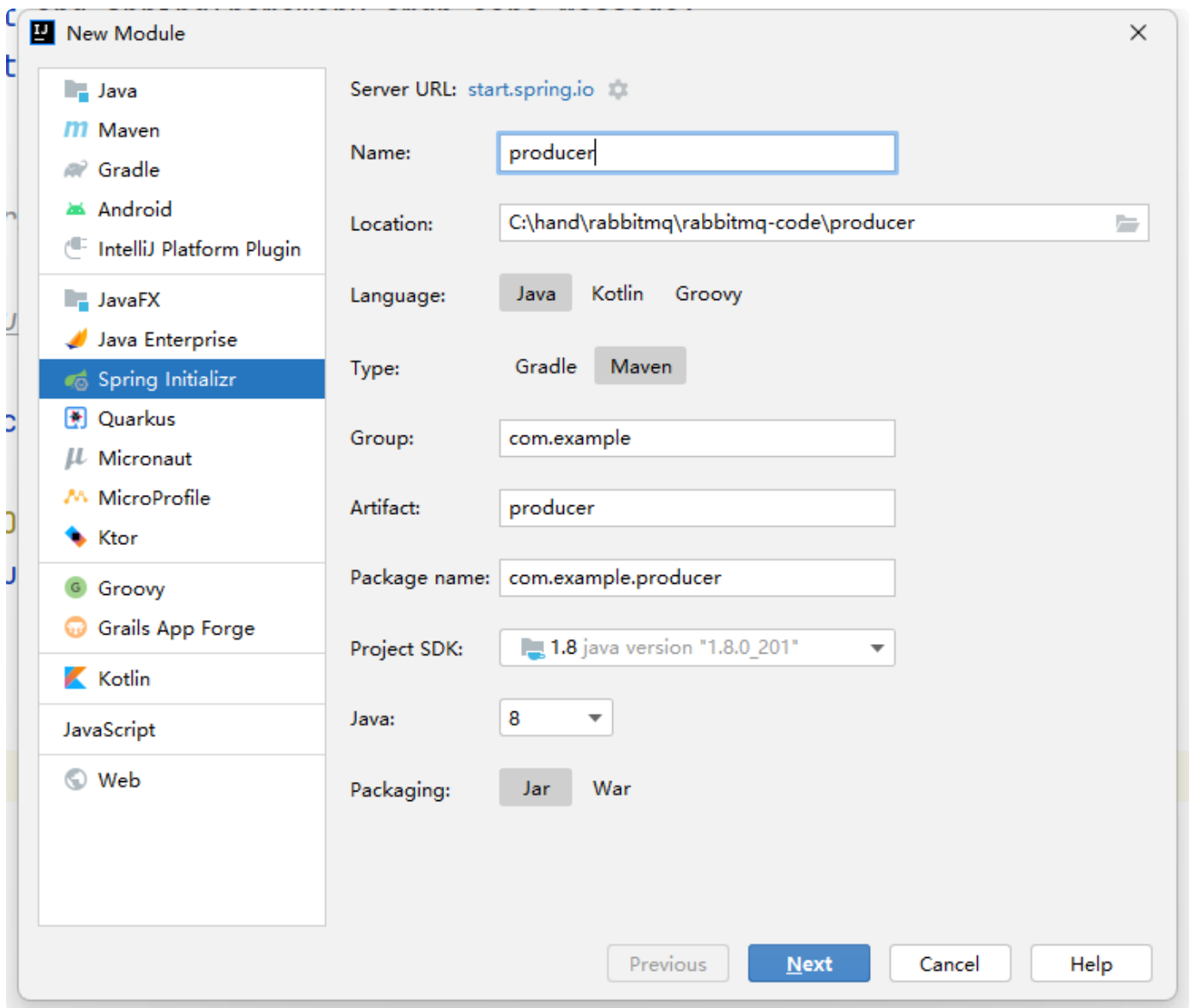


SpringBoot整合RabbitMQ

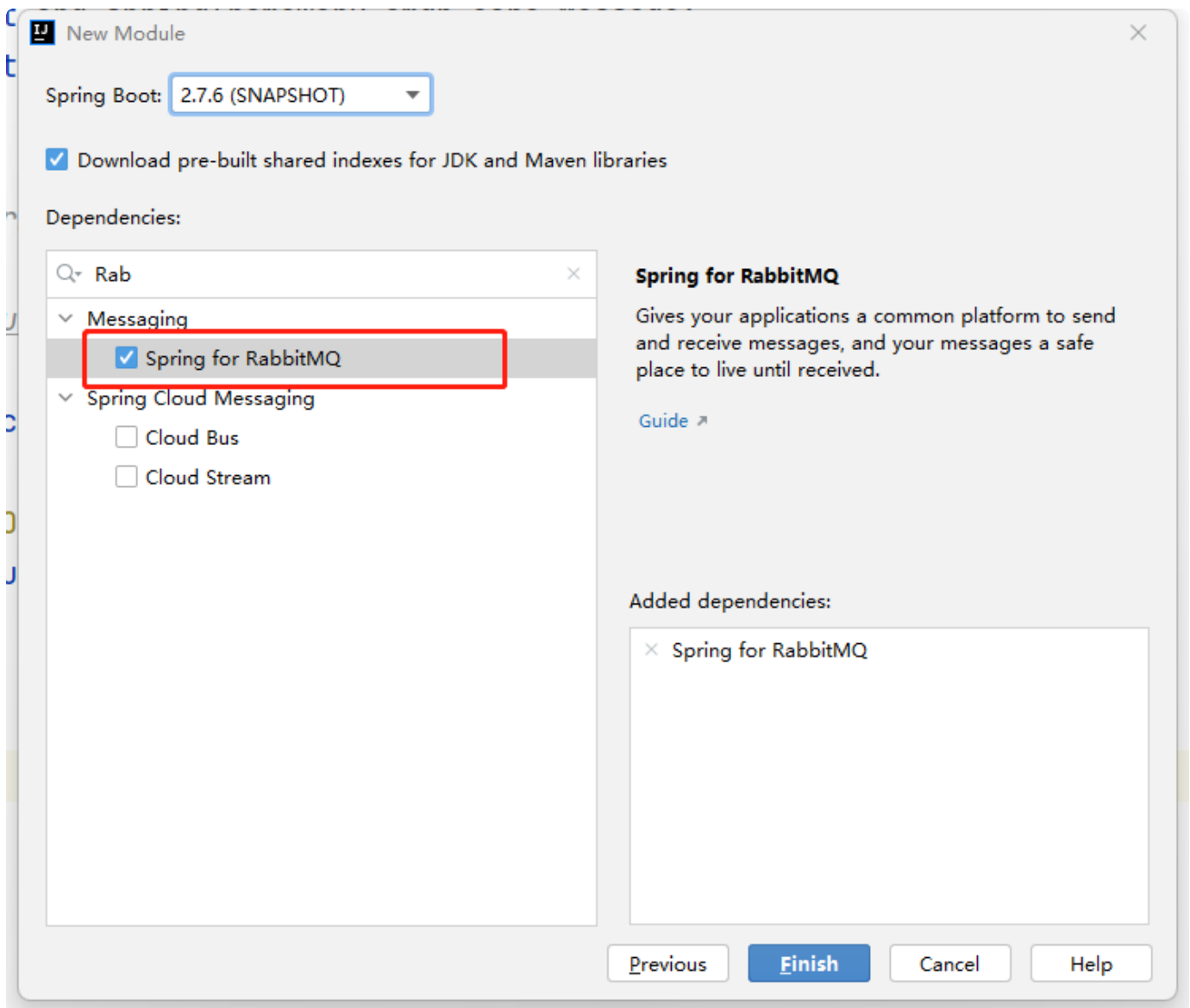
- 1.创建好消费者模块和生产者模块
 - 2.两个模块配置yml文件
 - 3.编写Producer-配置类
 - 4.编写Producer-测试类发送消息
 - 5.编写Consumer-监听类组件接收消息
- 总结

1.创建好消费者模块和生产者模块

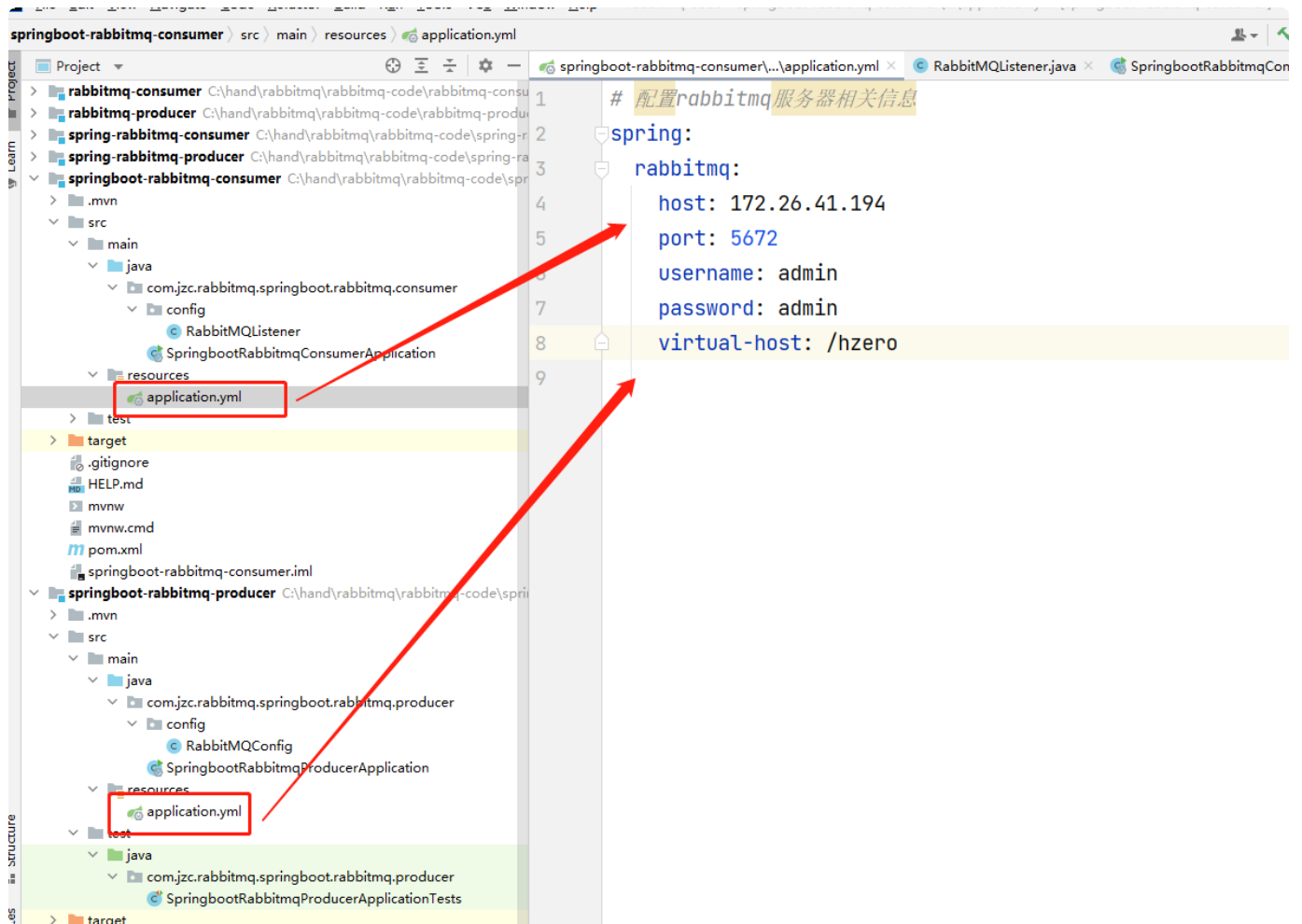
- 选择对应的Java版本和Maven库



- 添加Springboot-RabbitMQ依赖包



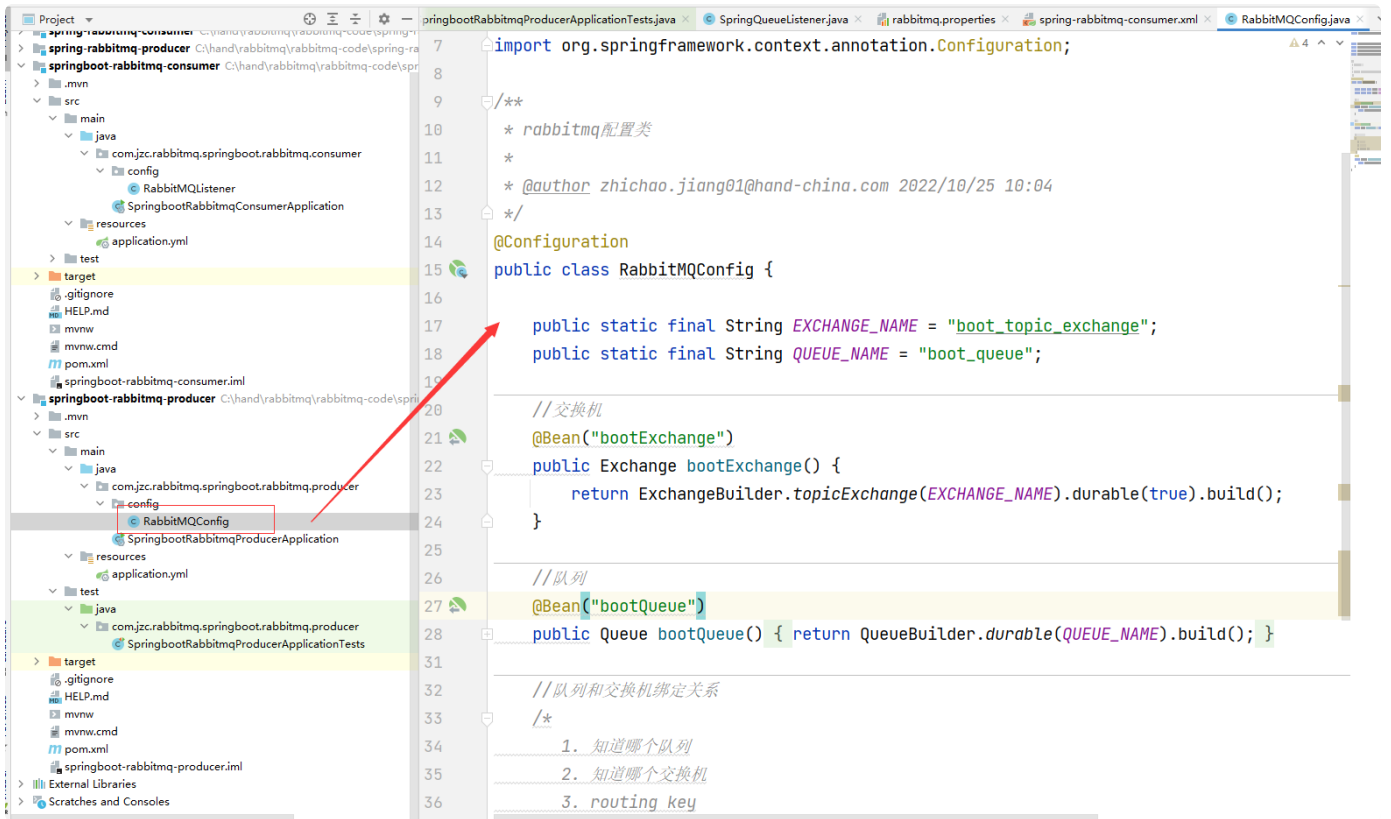
2.两个模块配置yml文件



YAML | 复制代码

```
1 # 配置rabbitmq服务器相关信息
2 spring:
3 rabbitmq:
4 host: 172.26.41.194
5 port: 5672
6 username: admin
7 password: admin
8 virtual-host: /hzero
```

3.编写Producer-配置类



- 利用ExchangeBuilder、QueueBuilder、BindingBuilder构建交换机、队列、绑定

```
1  /**
2   * rabbitmq配置类
3   *
4   * @author zhichao.jiang01@hand-china.com 2022/10/25 10:04
5   */
6   @Configuration
7   public class RabbitMQConfig {
8
9       public static final String EXCHANGE_NAME = "boot_topic_exchange";
10      public static final String QUEUE_NAME = "boot_queue";
11
12      //交换机
13      @Bean("bootExchange")
14      public Exchange bootExchange() {
15          return ExchangeBuilder.topicExchange(EXCHANGE_NAME).durable(true).build();
16      }
17
18      //队列
19      @Bean("bootQueue")
20      public Queue bootQueue(){
21          return QueueBuilder.durable(QUEUE_NAME).build();
22      }
23
24      //队列和交换机绑定关系
25      /**
26       * 1. 知道哪个队列
27       * 2. 知道哪个交换机
28       * 3. routing key
29       */
30      @Bean
31      public Binding bindQueueExchange(@Qualifier("bootQueue") Queue queue, @Qualifier("bootExchange") Exchange exchange){
32          return BindingBuilder.bind(queue).to(exchange).with("boot.#").noargs();
33      }
34  }
35
```

4.编写Producer–测试类发送消息

- 注入RabbitTemplate组件通过convertAndSend发送消息到对应的交换机

```
1 @SpringBootTest
2 @RunWith(SpringRunner.class)
3 class SpringbootRabbitmqProducerApplicationTests {
4
5     @Autowired
6     private RabbitTemplate rabbitTemplate;
7
8     @Test
9     void testSend() {
10         rabbitTemplate.convertAndSend(RabbitMQConfig.EXCHANGE_NAME,"boot.h
11         h","hello springboot rabbitmq");
12     }
13 }
14
```

Queues

All queues (15)

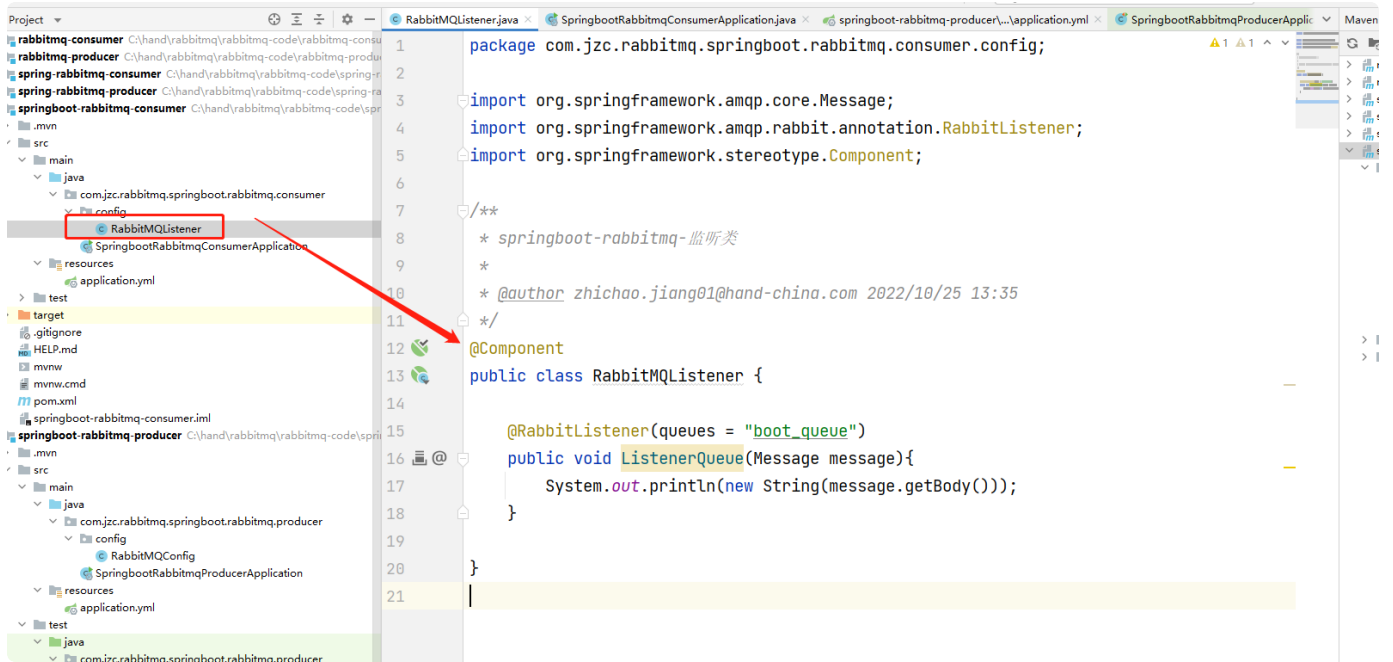
Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Overview					Messages			Message rates				+/-
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	Incoming	deliver	/ get	ack	
/hzero	boot_queue	classic	D	idle	1	0	1	0.00/s		0.00/s	0.00/s	
/hzero	hello_world	classic	D	idle	0	0	0					
/hzero	spring_fanout_queue_1	classic	D	idle	1	0	1	0.00/s				
/hzero	spring_fanout_queue_2	classic	D	idle	1	0	1	0.00/s				
/hzero	spring_queue	classic	D	idle	0	0	0	0.00/s		0.00/s	0.00/s	
/hzero	spring_topic_queue_star	classic	D	idle	0	0	0					
/hzero	spring_topic_queue_well	classic	D	idle	1	0	1	0.00/s				
/hzero	spring_topic_queue_well2	classic	D	idle	0	0	0					
/hzero	test_direct_queue1	classic	D	idle	0	0	0	0.00/s		0.00/s	0.00/s	
/hzero	test_direct_queue2	classic	D	idle	0	0	0	0.00/s		0.00/s	0.00/s	
/hzero	test_fanout_queue1	classic	D	idle	0	0	0	0.00/s		0.00/s	0.00/s	
/hzero	test_fanout_queue2	classic	D	idle	0	0	0	0.00/s		0.00/s	0.00/s	
/hzero	test_topic_queue1	classic	D	idle	0	0	0	0.00/s		0.00/s	0.00/s	
/hzero	test_topic_queue2	classic	D	idle	0	0	0	0.00/s		0.00/s	0.00/s	
/hzero	workQueue	classic	D	idle	0	0	0	0.00/s		0.00/s	0.00/s	

Add a new queue

5.编写Consumer-监听类组件接收消息



- @RabbitListener绑定对应的队列

```
Java | 复制代码

1  /**
2   * springboot-rabbitmq-监听类
3   *
4   * @author zhichao.jiang01@hand-china.com 2022/10/25 13:35
5   */
6   @Component
7   public class RabbitMQListener {
8
9       @RabbitListener(queues = "boot_queue")
10      public void listenerQueue(Message message){
11          System.out.println(new String(message.getBody()));
12      }
13
14  }
```



```
"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe" ...
```

[illegible]

```
2022-10-25 13:46:00.039 INFO 9464 --- [main] .c.SpringbootRabbitmqConsumerApplication : Starting SpringbootRabbitmqConsumerApplication using Java 1.8
2022-10-25 13:46:00.039 INFO 9464 --- [main] .c.SpringbootRabbitmqConsumerApplication : No active profile set, falling back to 1 default profile: "default"
2022-10-25 13:46:01.001 INFO 9464 --- [main] o.s.a.r.c.CachingConnectionFactory : Attempting to connect to: [172.26.41.194:5672]
2022-10-25 13:46:01.029 INFO 9464 --- [main] o.s.a.r.c.CachingConnectionFactory : Created new connection: rabbitConnectionFactory#69637b10:0/Si
2022-10-25 13:46:01.061 INFO 9464 --- [main] .c.SpringbootRabbitmqConsumerApplication : Started SpringbootRabbitmqConsumerApplication in 1.397 second
```

总结

SpringBoot提供了快速整合RabbitMQ的方式

基本信息再yml中配置，队列交互机以及绑定关系在配置类中使用Bean的方式配置

生产端直接注入RabbitTemplate完成消息发送

消费端直接使用@RabbitListener完成消息接收