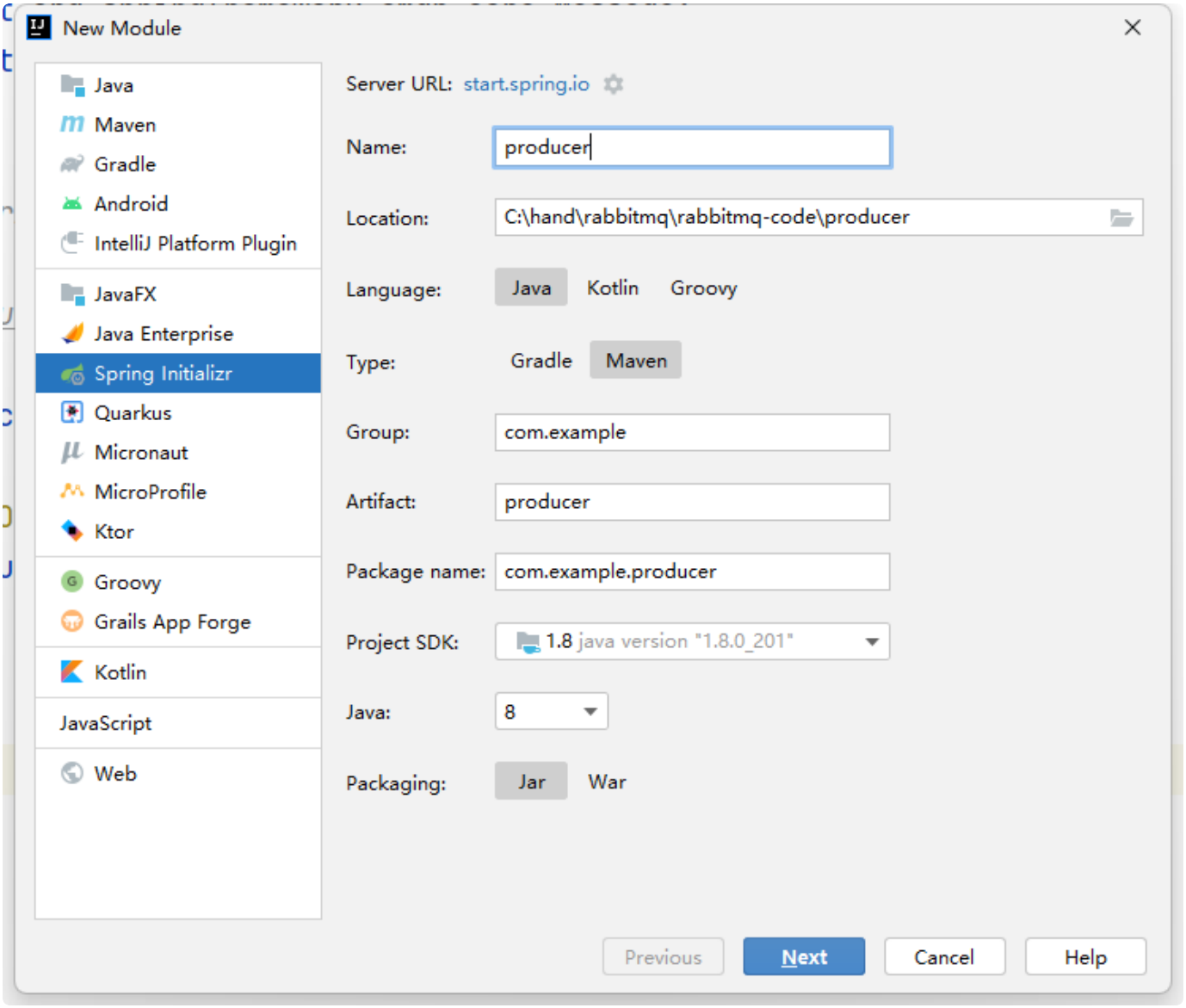
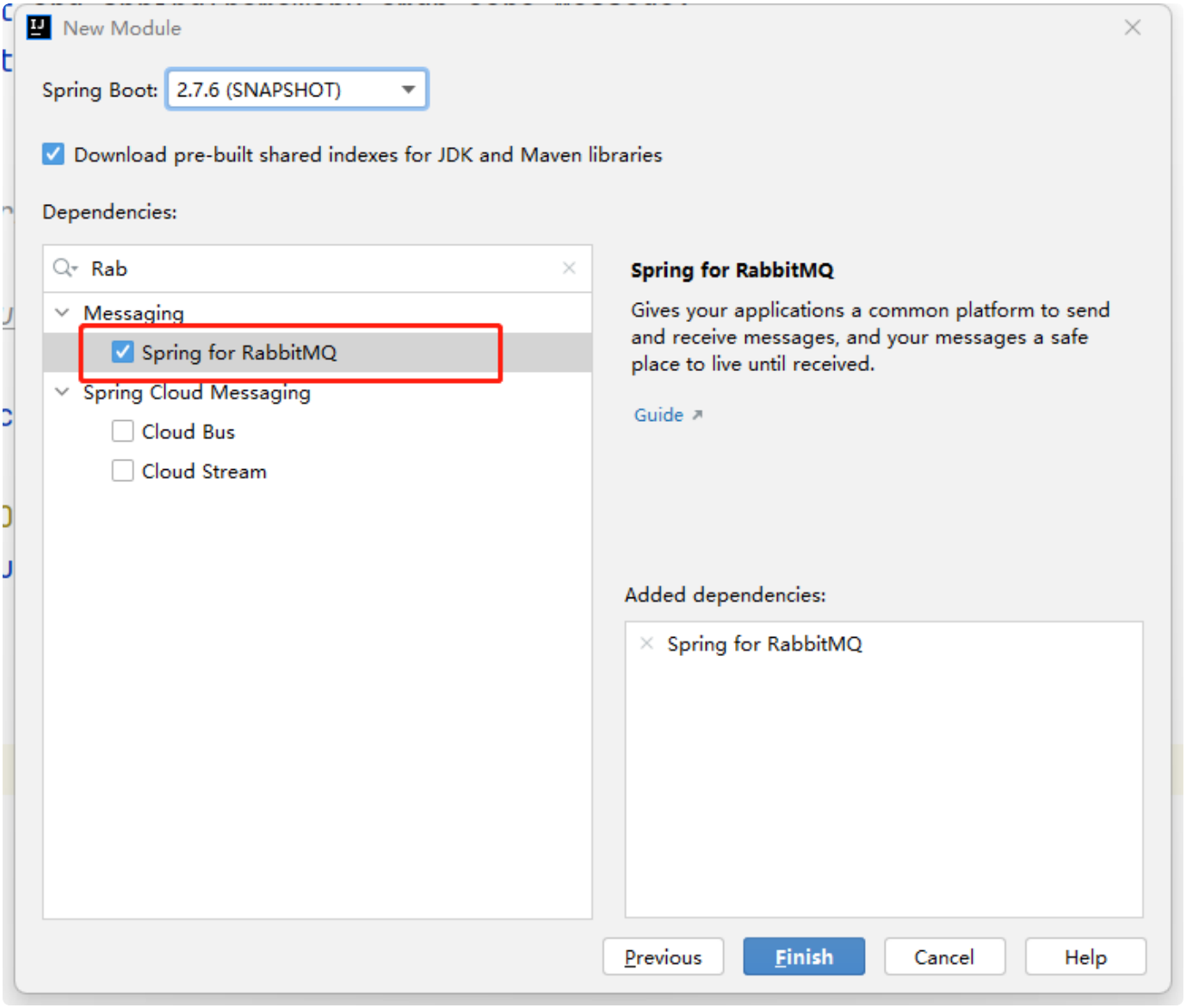
SpringBoot整合RabbitMQ

1.创建好消费者模块和生产者模块

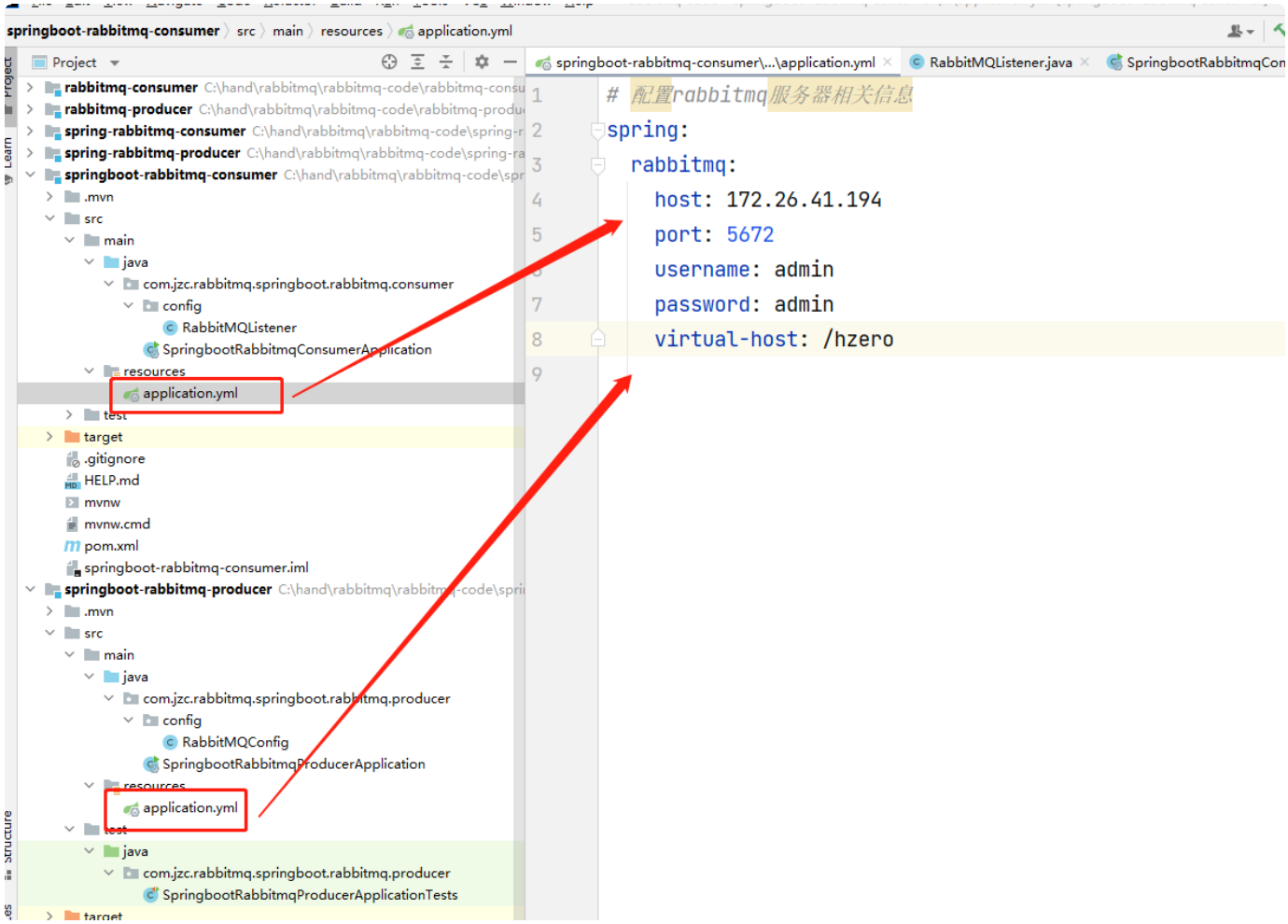
- 选择对应的Java版本和Maven库



- 添加Springboot–RabbitMQ依赖包



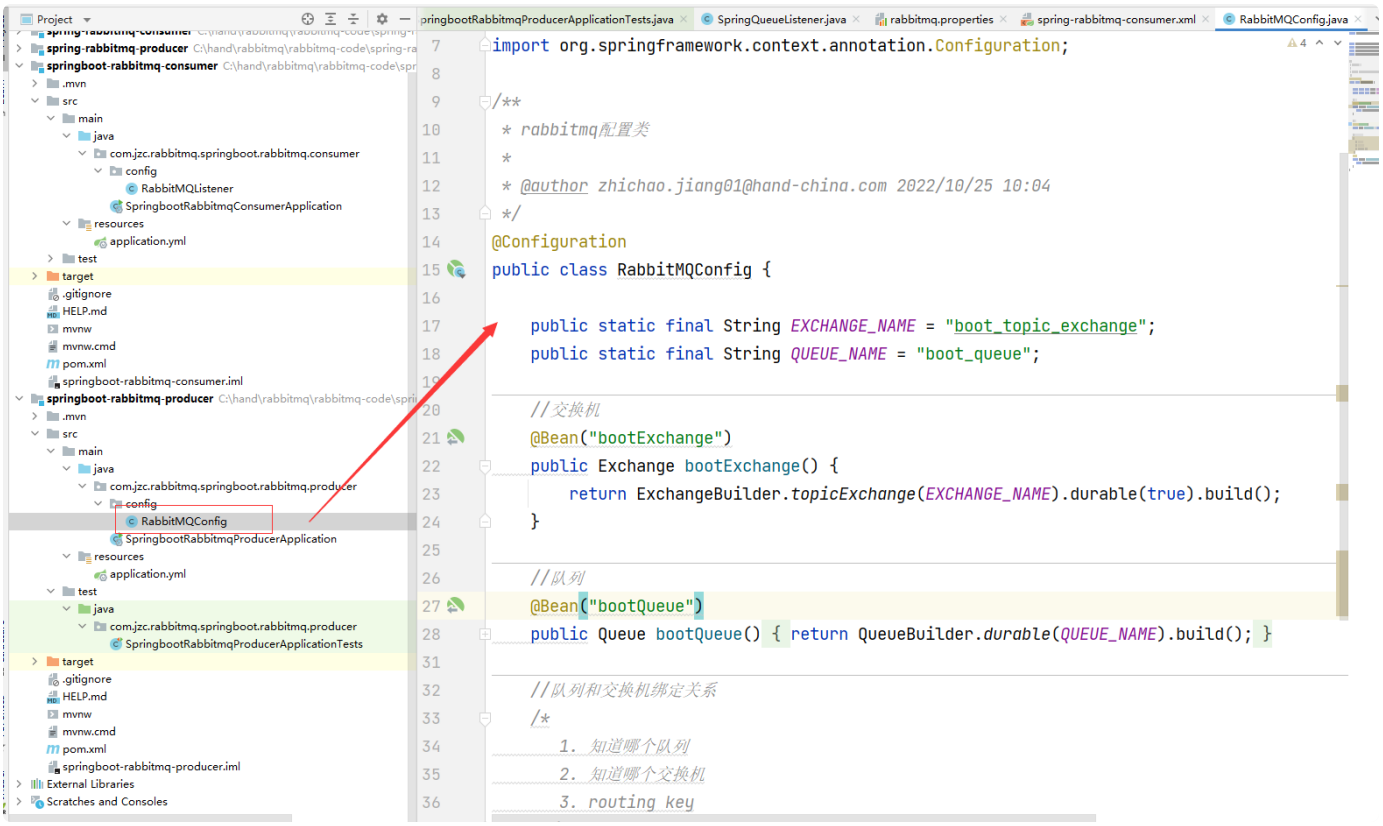
2.两个模块配置yml文件



YAML | 复制代码

```yaml
# 配置rabbitmq服务器相关信息
spring:
rabbitmq:
host: 172.26.41.194
port: 5672
username: admin
password: admin
virtual-host: /hzero
```

3.编写Producer-配置类



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

Java | 复制代码

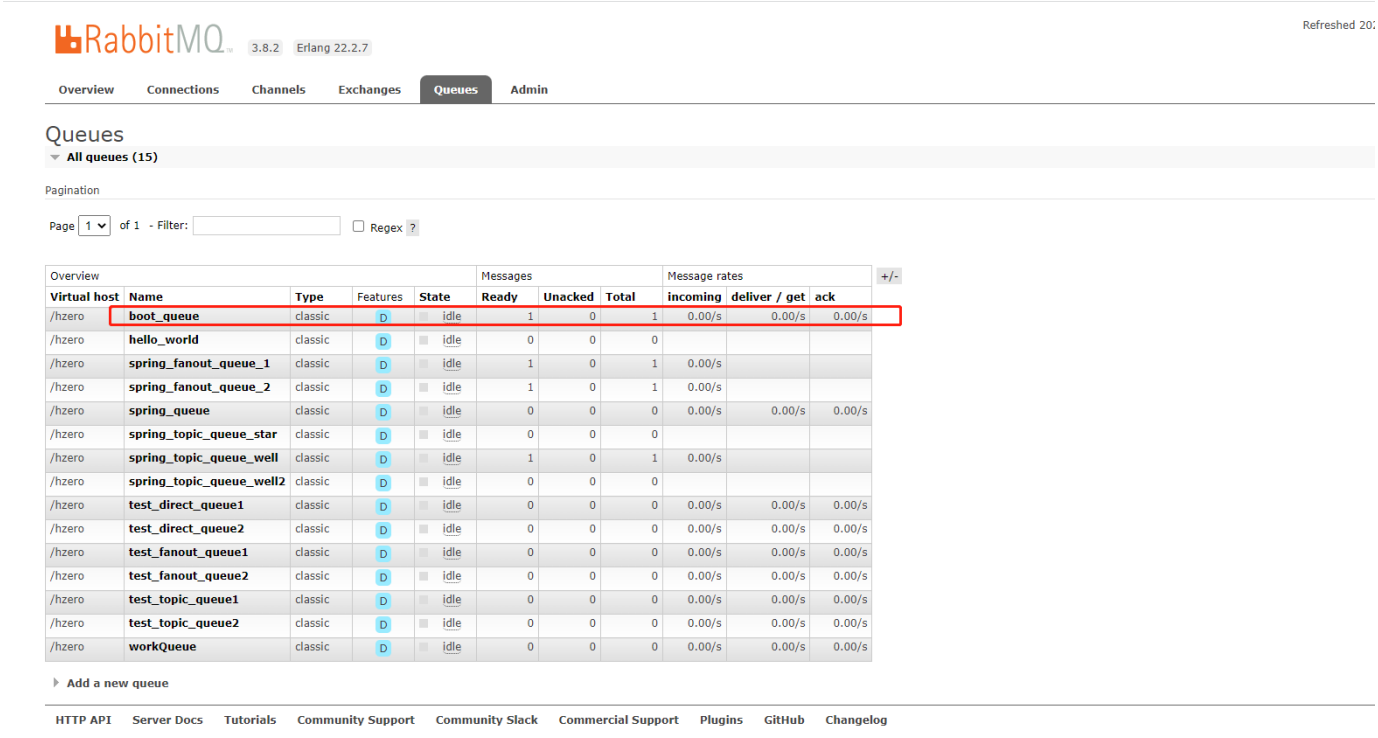
```java
/**
* rabbitmq配置类
*
* @author zhichao.jiang01@hand-china.com 2022/10/25 10:04
*/
@Configuration
    public class RabbitMQConfig {

        public static final String EXCHANGE_NAME = "boot_topic_exchange";
        public static final String QUEUE_NAME = "boot_queue";

        //交换机
        @Bean("bootExchange")
        public Exchange bootExchange() {
            return ExchangeBuilder.topicExchange(EXCHANGE_NAME).durable(true).build();
        }

        //队列
        @Bean("bootQueue")
        public Queue bootQueue(){
            return QueueBuilder.durable(QUEUE_NAME).build();
        }

        //队列和交换机绑定关系
        /*
          1. 知道哪个队列
          2. 知道哪个交换机
          3. routing key
        */
        @Bean
        public Binding bindQueueExchange(@Qualifier("bootQueue") Queue queue,@Qualifier("bootExchange") Exchange exchange){
            return BindingBuilder.bind(queue).to(exchange).with("boot.#").noargs();
        }
    }

```

4.编写Producer-测试类发送消息

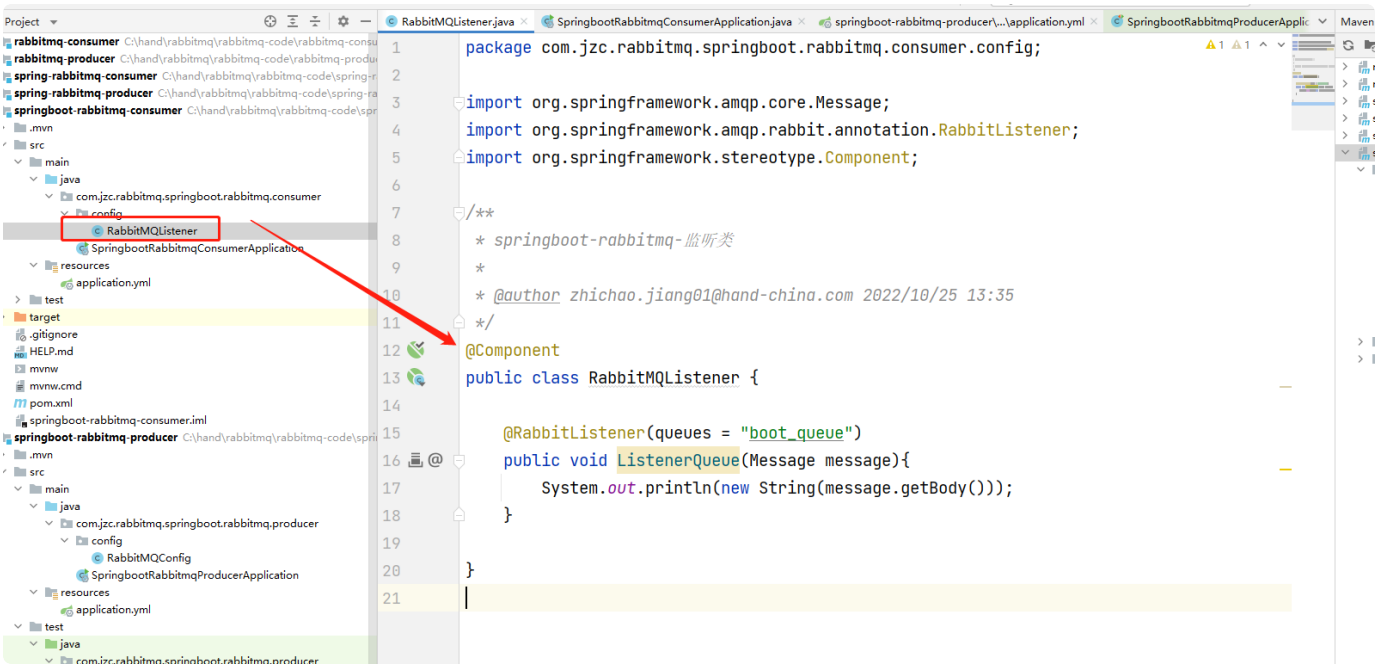
- 注入RabbitTemplate组件通过convertAndSend发送消息到对应的交换机

Java 复制代码

```java
@SpringBootTest
@RunWith(SpringRunner.class)
class SpringbootRabbitmqProducerApplicationTests {

    @Autowired
    private RabbitTemplate rabbitTemplate;

    @Test
    void testSend() {
        rabbitTemplate.convertAndSend(RabbitMQConfig.EXCHANGE_NAME,"boot.hh","hello springboot rabbitmq");
    }

}

```



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

```java
/**
 * springboot-rabbitmq-监听类
 *
 * @author zhichao.jiang01@hand-china.com 2022/10/25 13:35
 */
@Component
public class RabbitMQListener {

    @RabbitListener(queues = "boot_queue")
    public void ListenerQueue(Message message){
        System.out.println(new String(message.getBody()));
    }

}
```

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

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::       (v2.7.6-SNAPSHOT)

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: "de
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
hello springboot rabbitmq
```

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