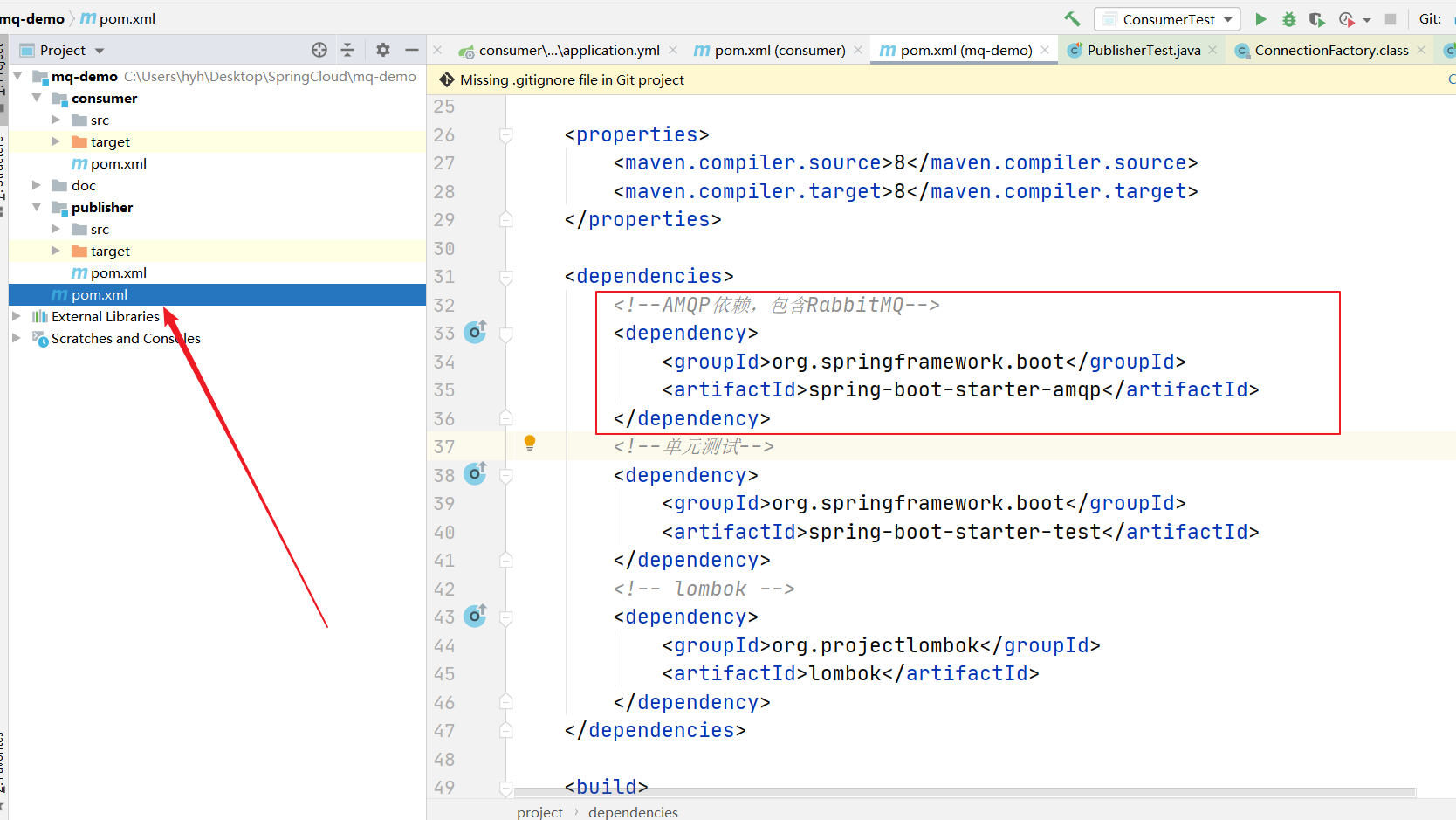
# SpringAMQP实现基本消息队列Basic Queue

## 消息发送

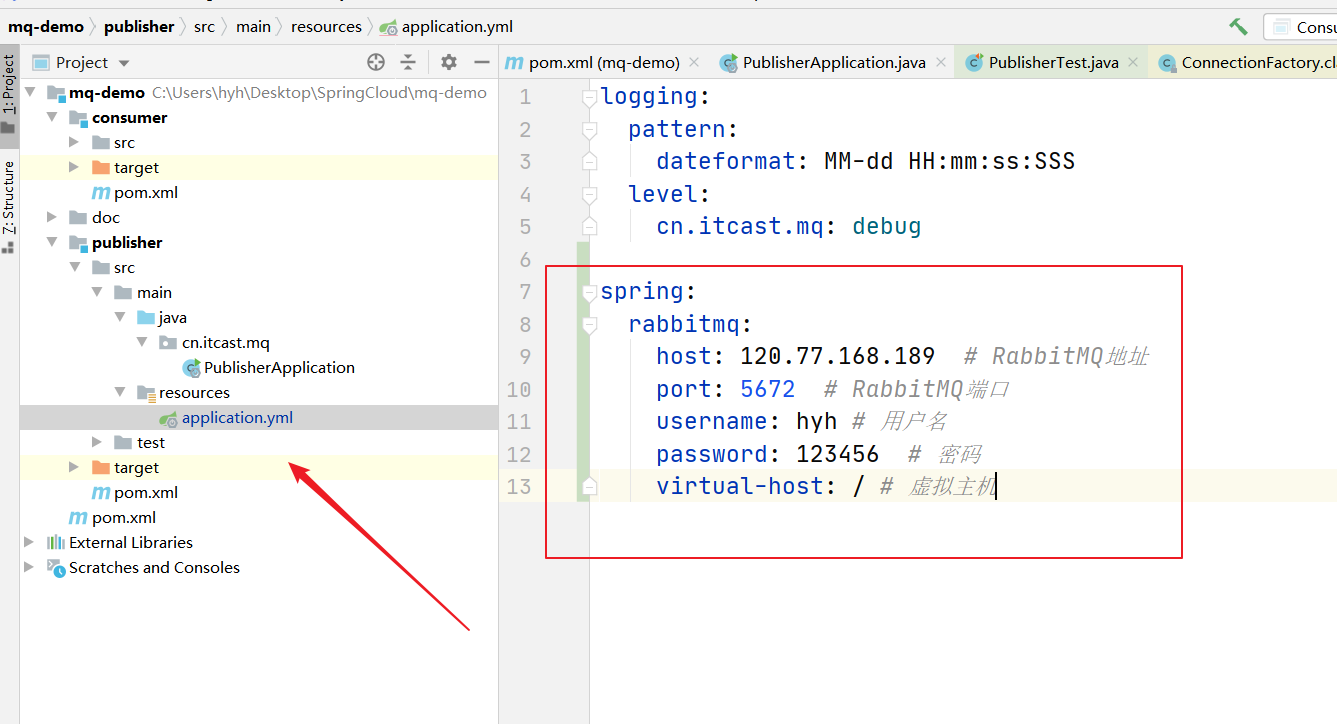
### 1；在父工程中引入spring-amqp的依赖

这里为什么要在夫工程引入，因为消息的发送和消息的接收都要依赖spring-amqp，即spring-amqp不仅仅publisher用到，在consumer中也要用到

可以看到没有引入spring-rabbit，因为spring-rabbit就是spring-amqp的默认实现!



### 2：在application.yml配置文件中配置RabbitMQ的信息：



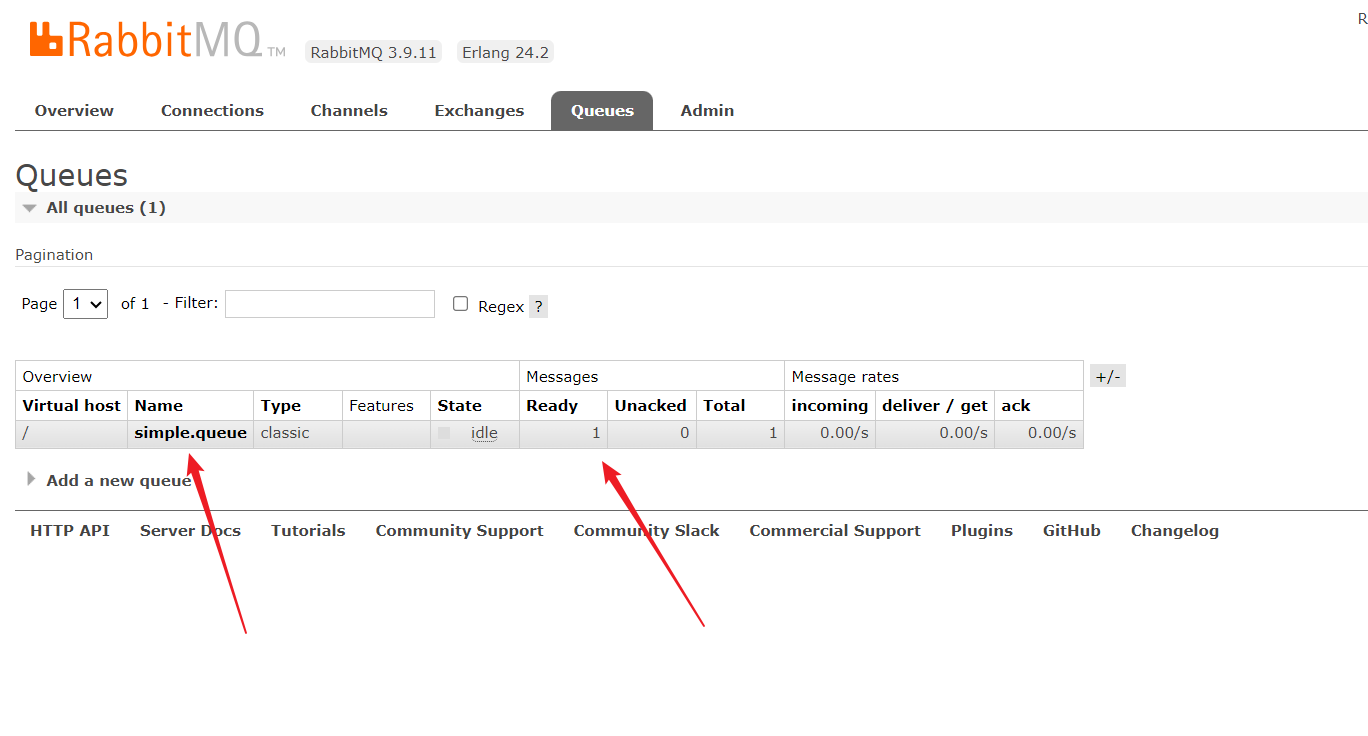
### 3：在publisher服务中新建一个测试类，编写测试方法

在publisher服务中利用RabbitTemplate发送消息到simple.queue这个队列，RabbitTemplate就是类似于Redis中的RedisTemplate，都是Spring帮我们提供、封装好的模板工具，用RabbitTemplate直接发消息就行了，贼简单！

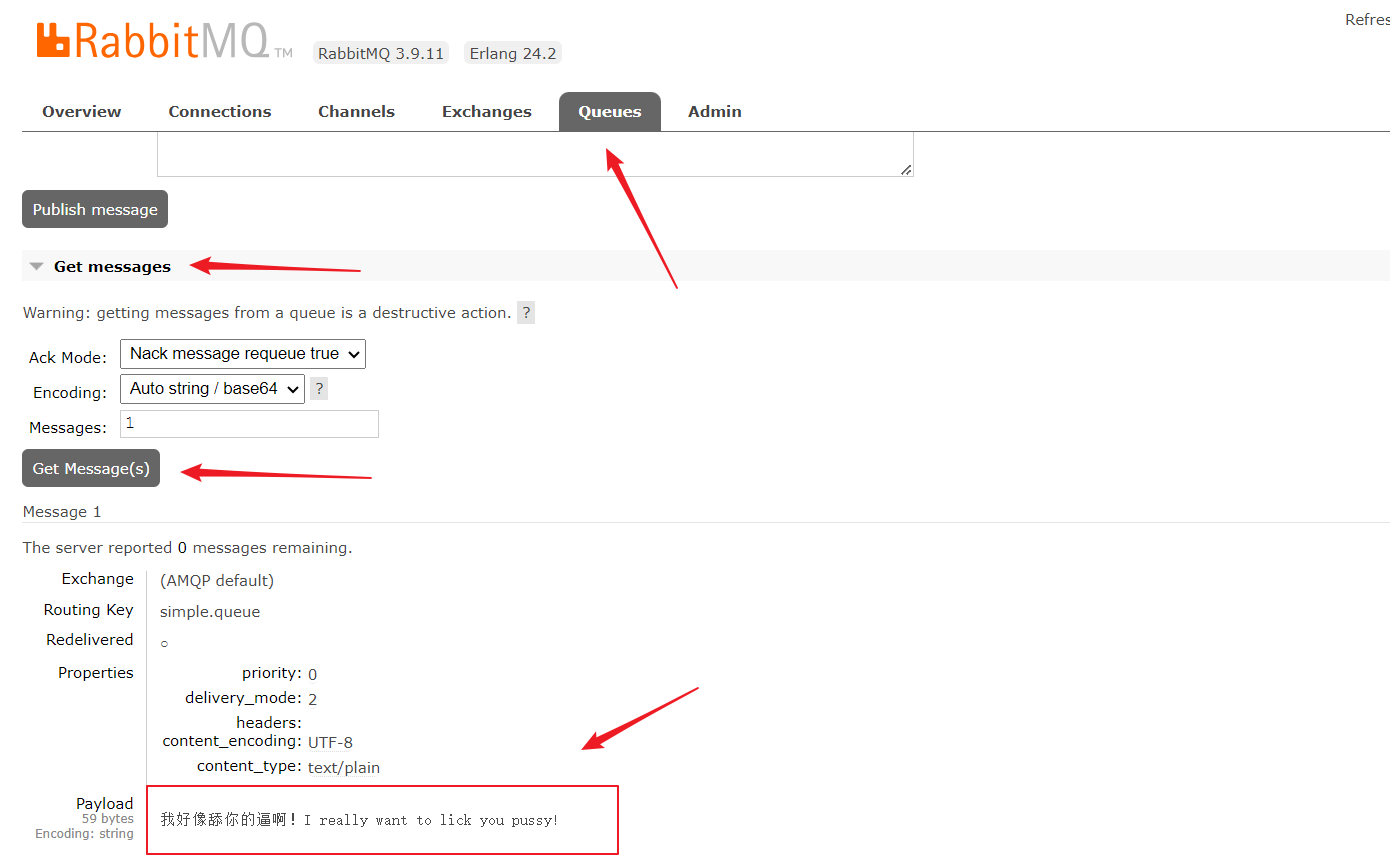
这里使用单元测试来做的，将来你坑定不是用单元测试来发消息，一定是在服务中做业务发消息，比如支付册成功后想消息，就是在支付之后的业务里加上发消息的代码就ok了！



### 4：运行测试方法，在RabbitMQ查看发送的消息，发现确实有一个消息发送过来了！



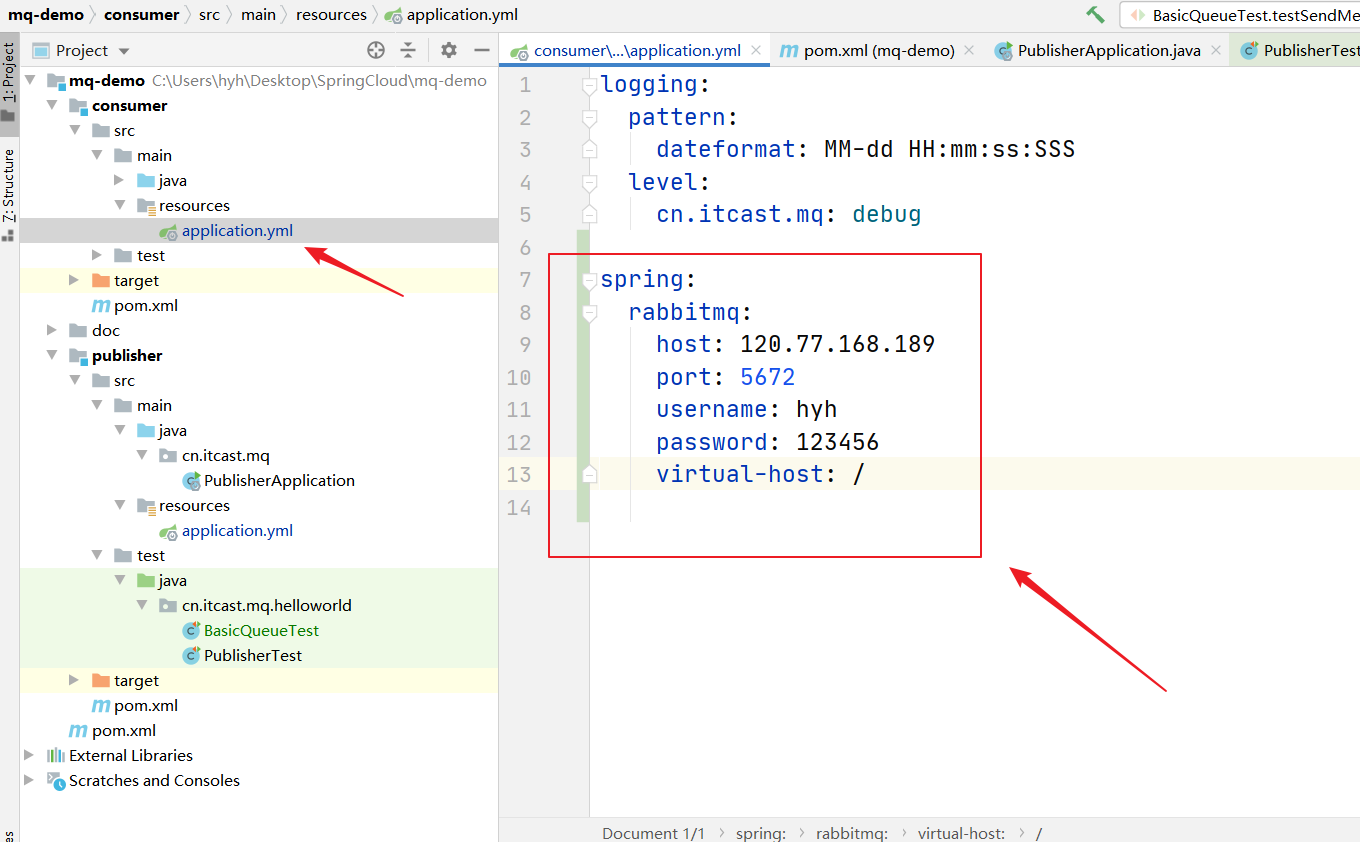
### 5：查看消息详细内容，没错就是我们发送的消息



## 消息接收、消费

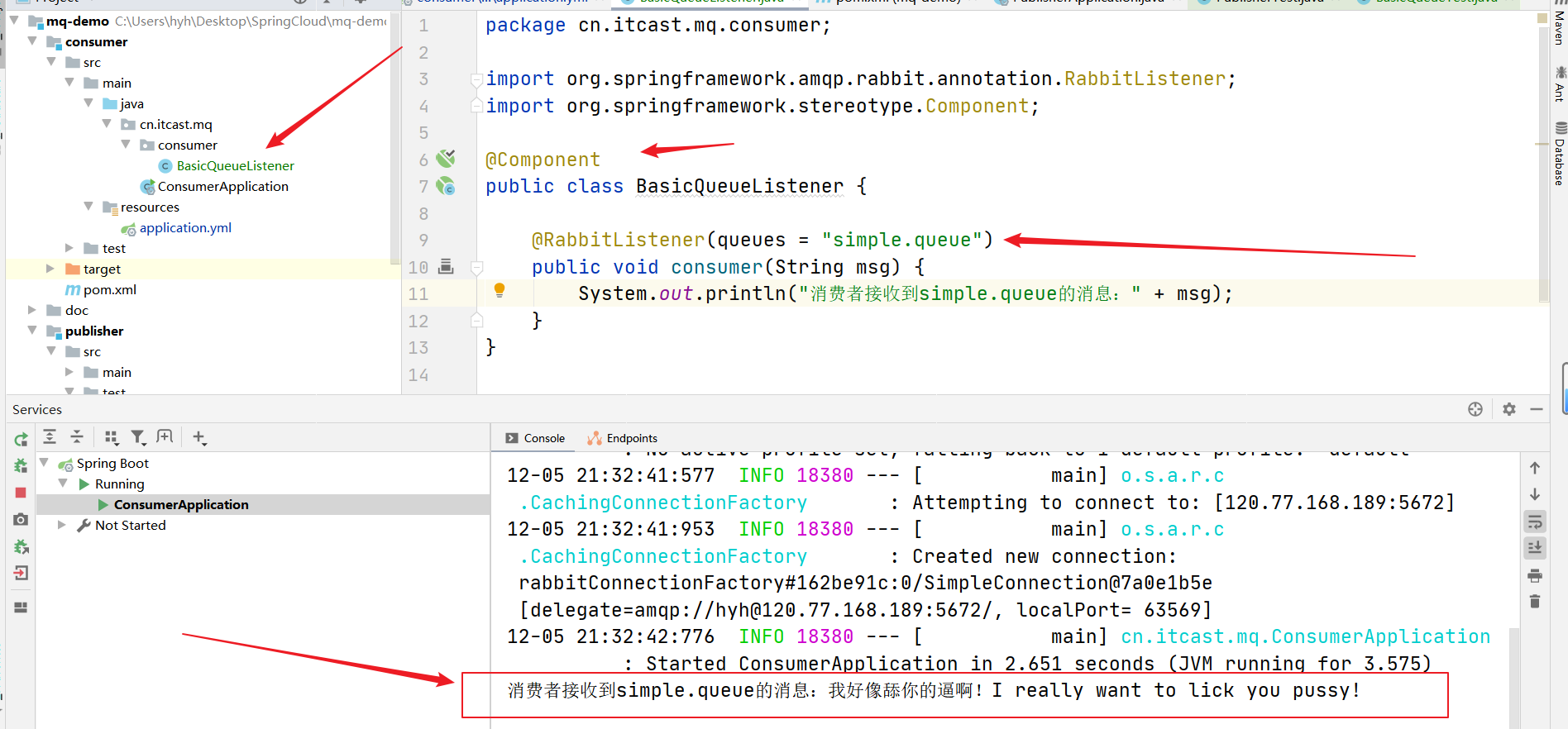
在consumer服务中编写消费逻辑，绑定simple.queue这个队列

### 1：在配置文件application.yml中配置RabbitMQ信息



### 2：定义监听消息的类和接收、消费消息的方法

接下来要做的事就和publisher不一样了，因为消息消费者要做的事是监听消息，Spring已经帮我们和RabbitMQ建立了连接（连接、通道），我们consumer唯一要做的事就是要监听什么队列！监听到队列有消息了要干什么事！在consumer服务中定义一个类，类种定义处理消息的行为方法，但是类要加@Component注解，打这个类声明成一个bean，Spring就能找到这个类了，然后再消息消费的方法上要加@RabblitListener注解，这个注解是用来声明队列的名称的，那将来Spring就知道了这个方法是消息消费方法，并且知道要监听什么队列，将来一旦有对应队列的消息Spring就能监听到，Spring就会立即投放到这个消息消费方法中，就能处理对应的消息了，就能对这个消息为所欲为了！相当优雅！



### 3：查看RabbitMQ管理后台，发现消息没有了

证明了我们之前的结论，RabbitMQ是阅后即焚，就是在队列的消息被接收消费了，就会在队列中移除！

