## SpringAMQP-消息转换器：

其实这个消息转换器，一致贯穿我们之前的学习！

### 说明：

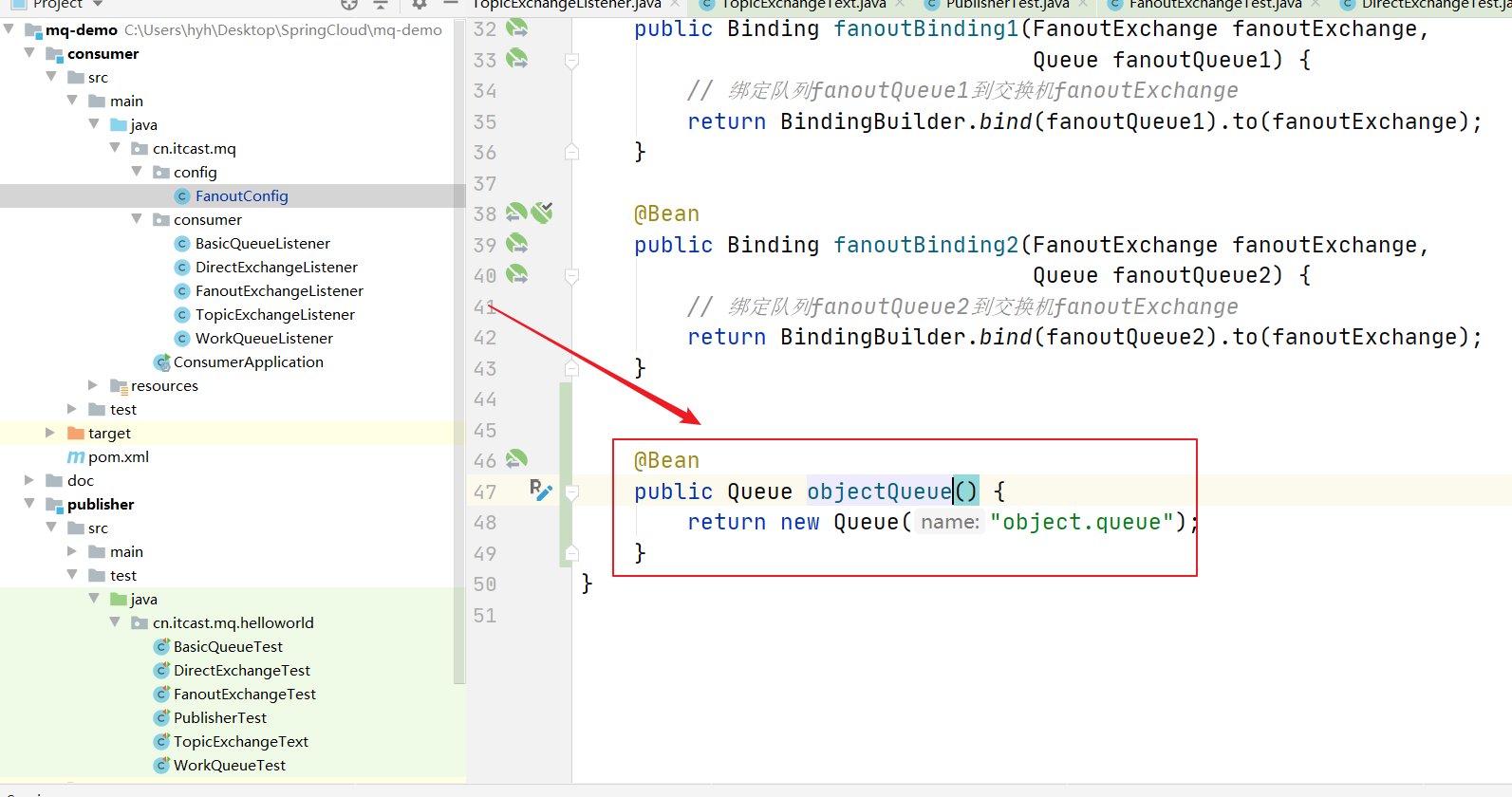
有没有注意到，在SpringAMQP的中，我们使用RabbitTemplate这个工具的时候发送消息时，方法参数列表中消息的类型一直是Object，如下图所示：



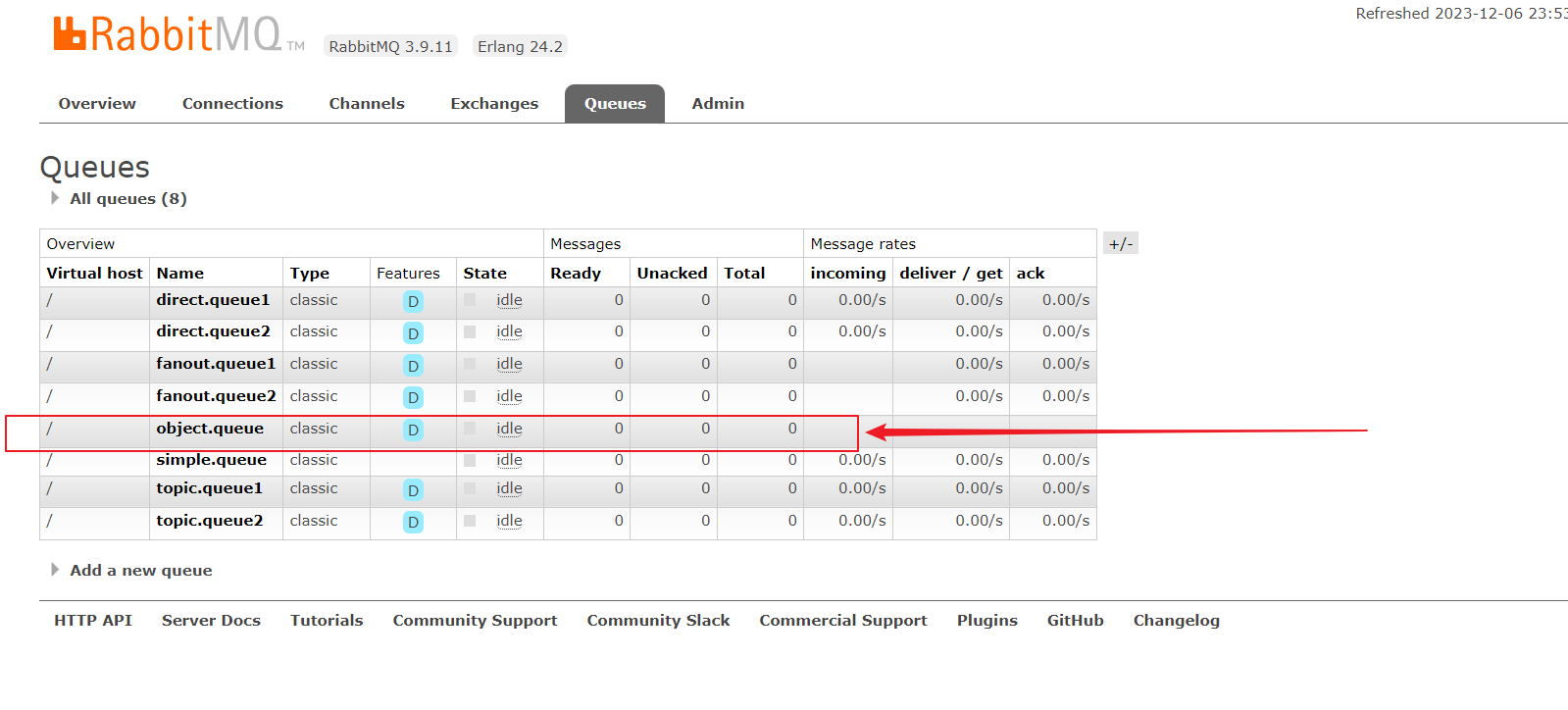
这说明什么，难道说SpringAMQP允许我们发的消息是任意类型的对象吗？我们可以验证一下：

### 1：在consumer中声明一个队列：object.queue，然后重启consumer。

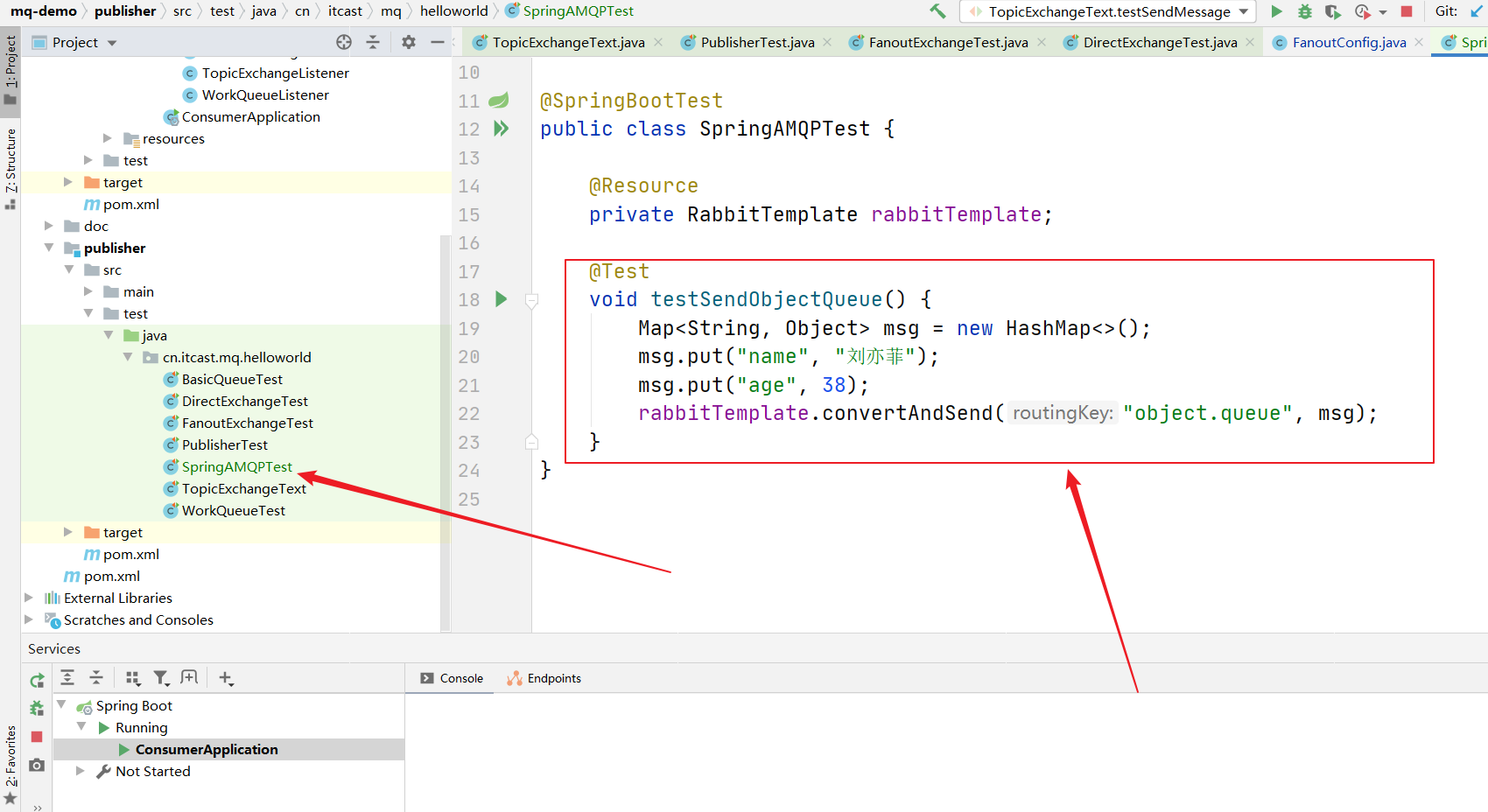
为什么不再@RabblitListener中声明，因为我想在RabbitMQ管理后台看看发送的对象在队列中什么样子，在@RabblitListener中声明，publisher一发送消息就被消费掉了



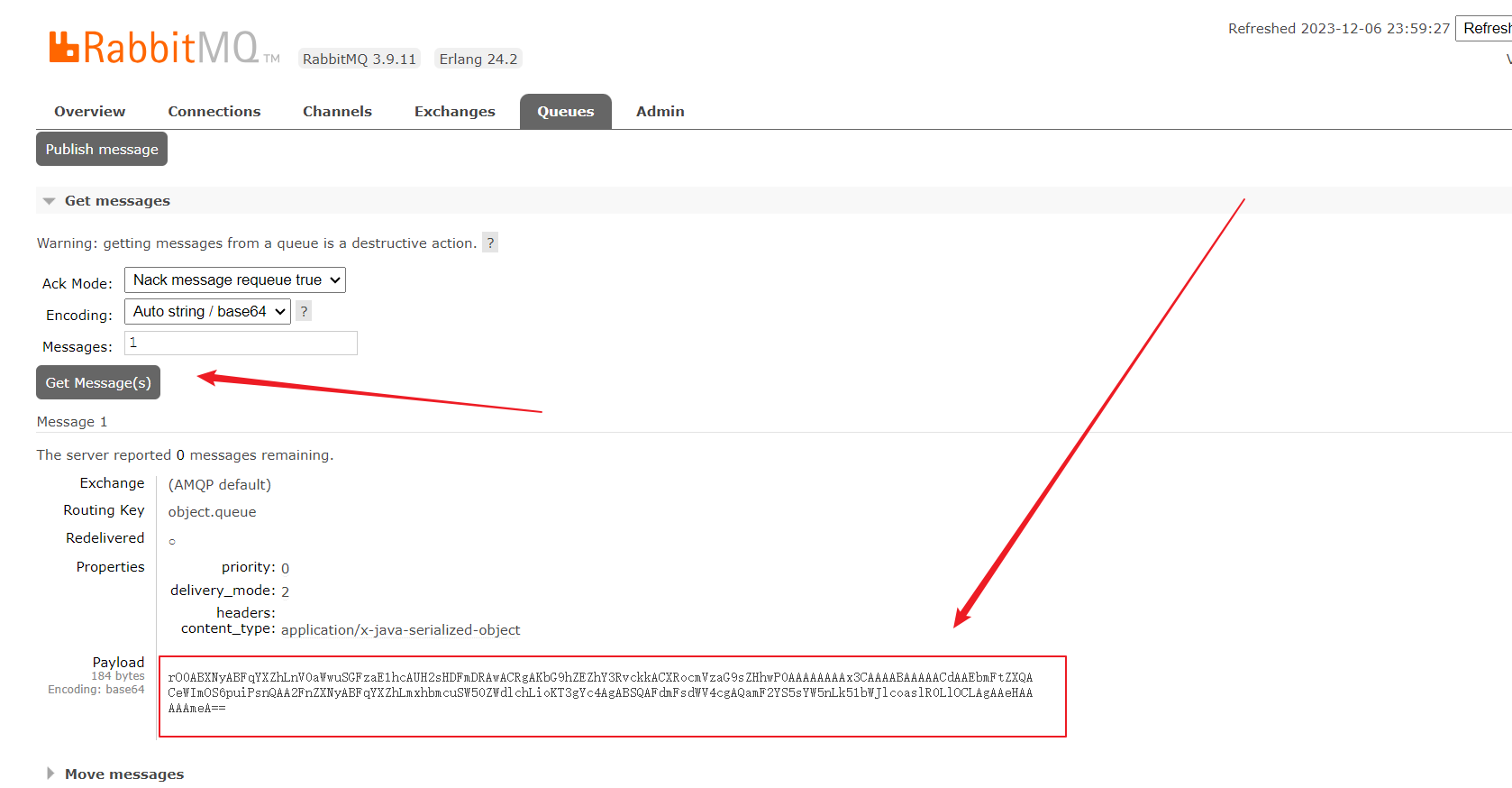
### 2：在RabbitMQ管理后台查看创建的队列object.queue



### 3：在publisher编写测试用例，往队列object.queue发送map类型对象



### 4：运行测试用例，在RabbitMQ查看消息详情



**你看，确实能发送任何类型的消息**，但是，在RabbitMQ管理后台的消息内容我们看不懂啊！

## 默认的消息对象序列化方式的问题

可以看到消息默认的类型是Java-serialized-object，java序列化，哦！因为我们的RabbitMQ默认只支持字节，我们RabbitMQ原生API的时候发的就只能是字节！而SpringAMQP却允许我们发对象，说明SpringAMQP会将我们的发的消息对象做序列化！用的序列化方式就是：Java-serialized-object，也就是JDK的序列化实现，这种序列化有缺点：一是性能比较差，二是安全性有问题，容易出现注入的问题；

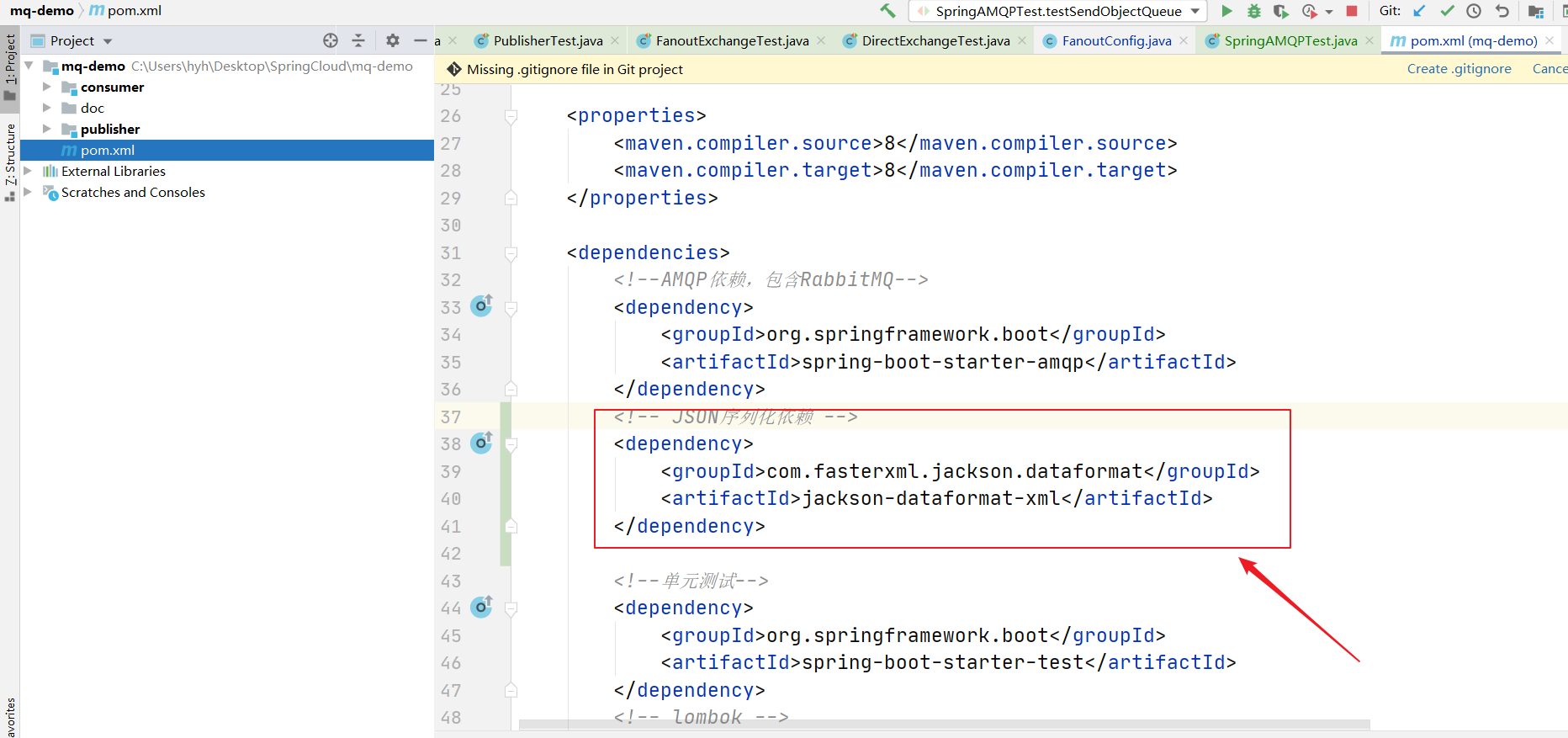
三是数据长度也太长了点把，见上面图中所指消息的内容，我明明只发了苗条的刘亦菲，却变成了这么长的一串东西，还我刘亦菲！而我们知道消息体越大，传输消息的速度也就越慢！而且还占用额外的内存空间。所以说，非常不推荐用这种默认的序列化方式！

### 问题解决

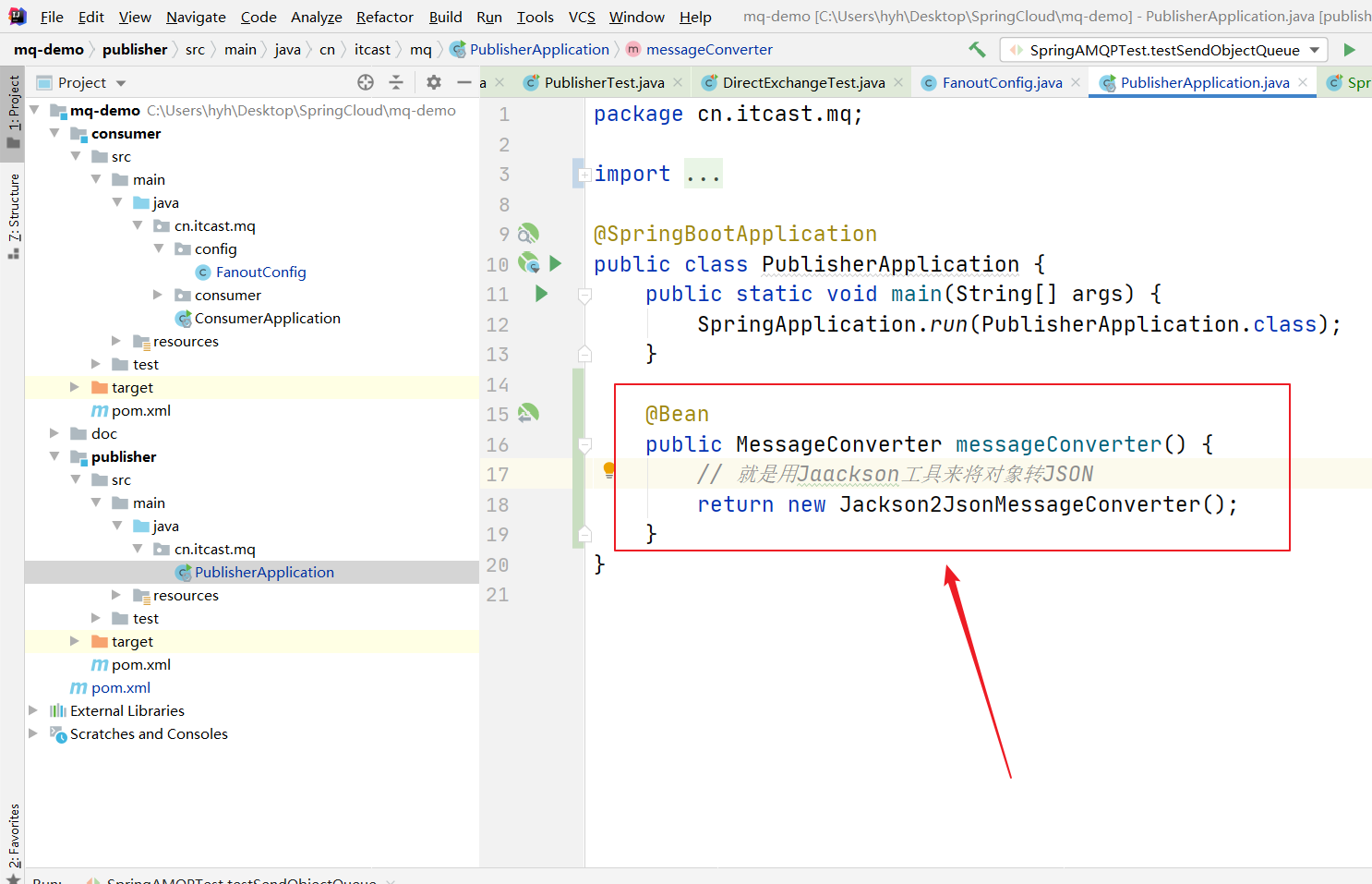
Spring的对消息对象的处理是由**MessageConverter**的类来处理的。而默认实现是SimpleMessageConverter，基于JDK的ObjectOutputStream完成序列化。如果要修改只需要定义一个MessageConverter类型的Bean即可，我们推荐用Json方式序列化，Json序列化方式消息体更加短小精悍，消息传输速度比默认的快！

步骤如下：

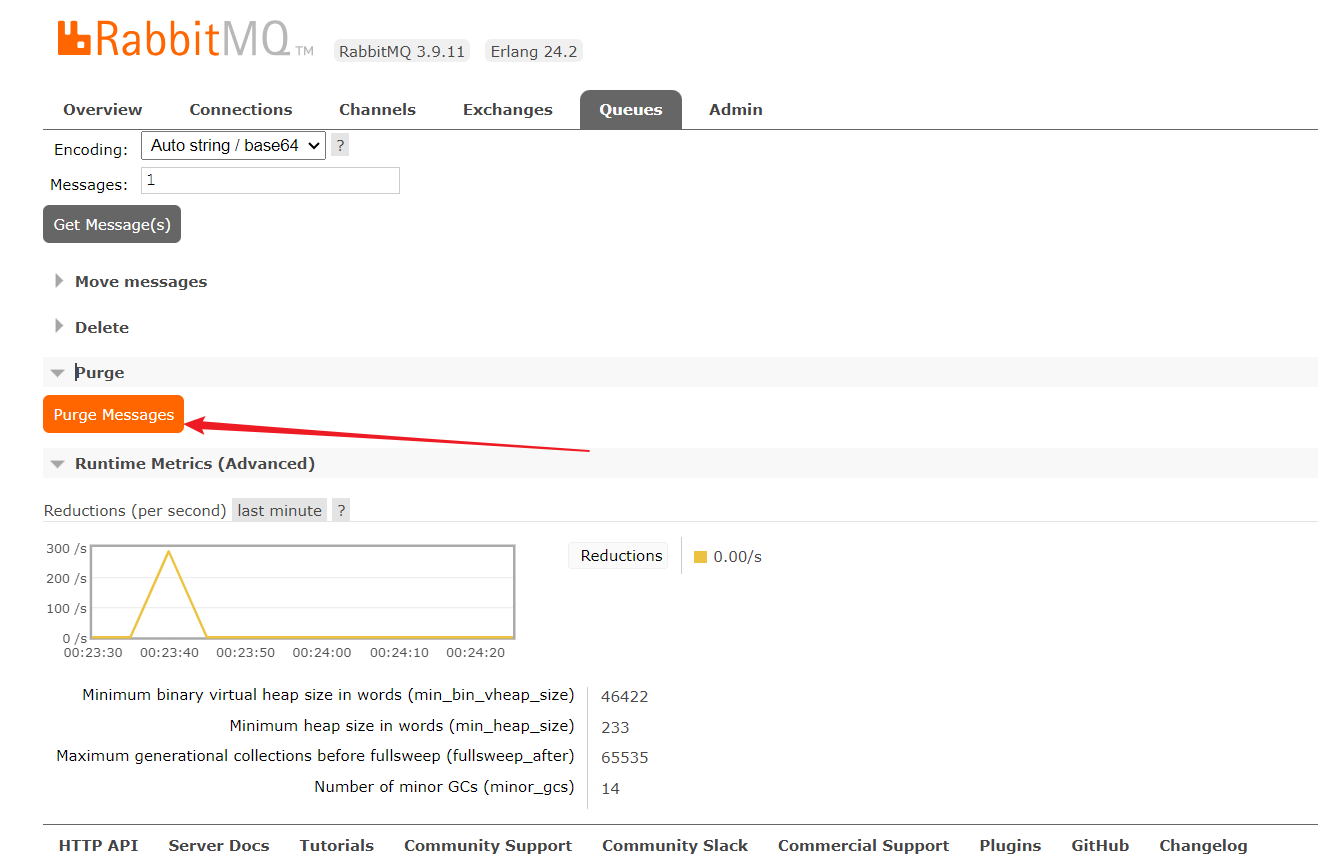
#### 1：在publisher服务、consumer服务引入jackson-dataformat-xml依赖，父工程引入即可：



#### 2：在publisher服务、consumer服务声明MessageConverter：Jackson2JsonMessageConverter

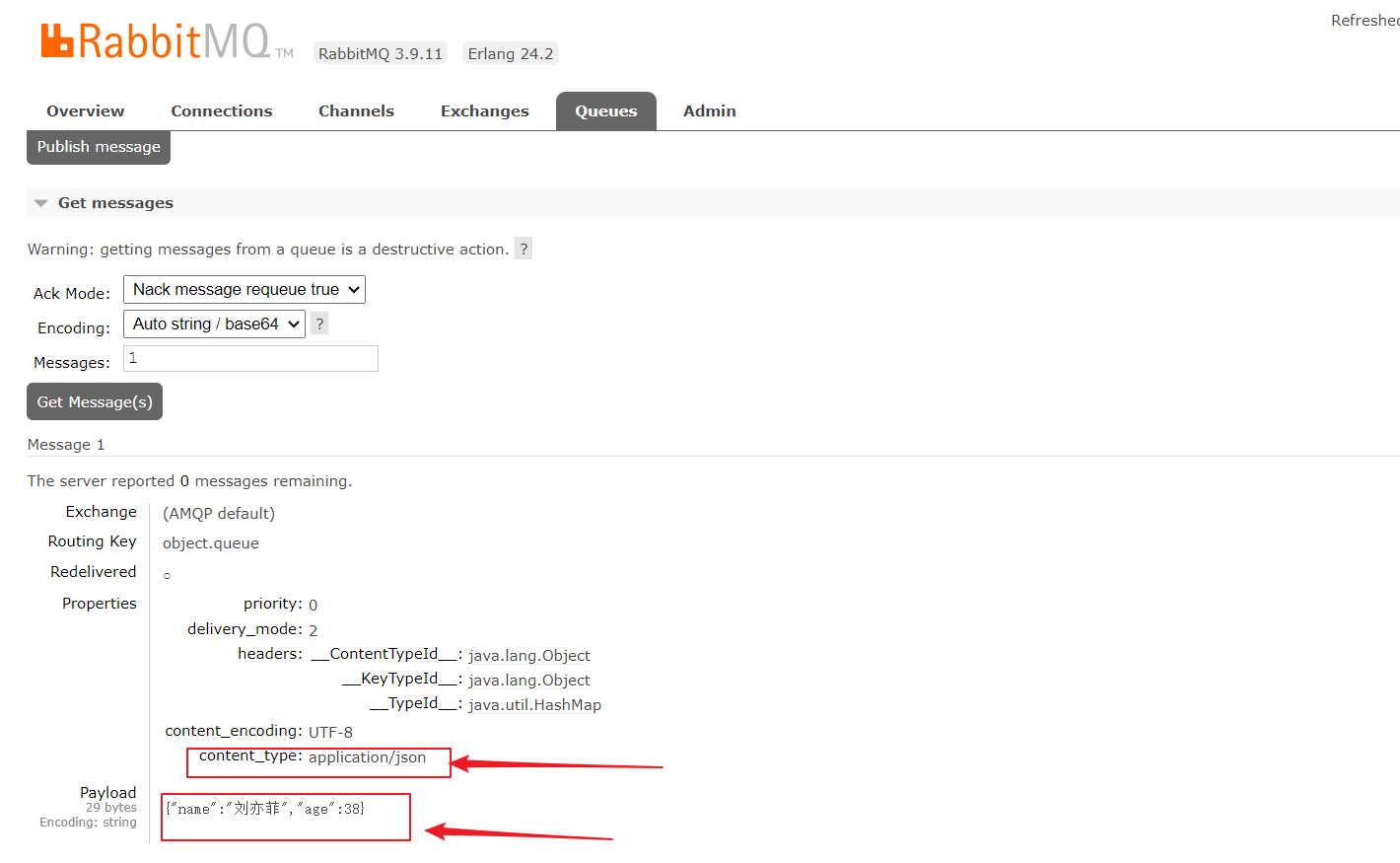


#### 3：先到RabbitMQ管理后台把就消息删除

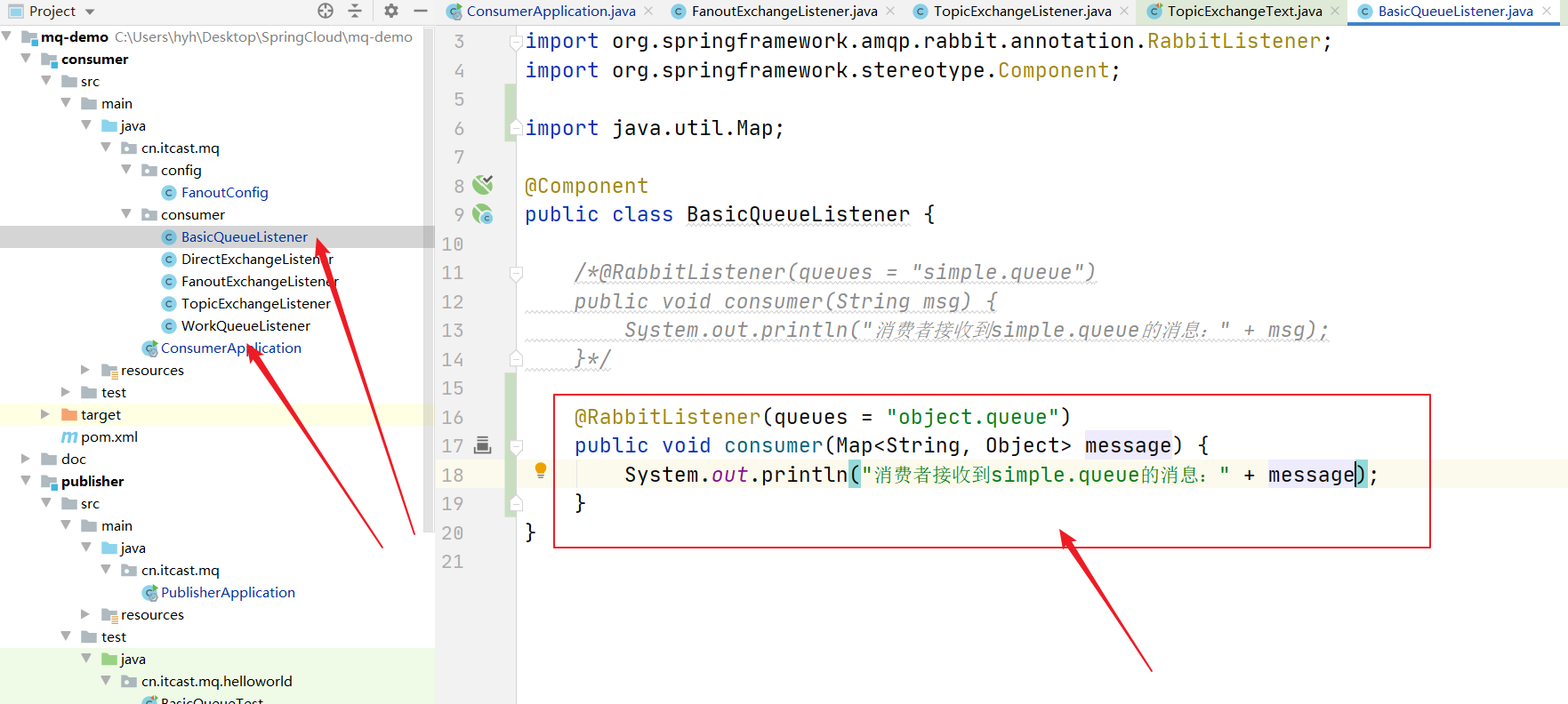


#### 4：运行测试用例方法发送消息刘亦菲，然后再RabbitMQ管理后台查看消息内容

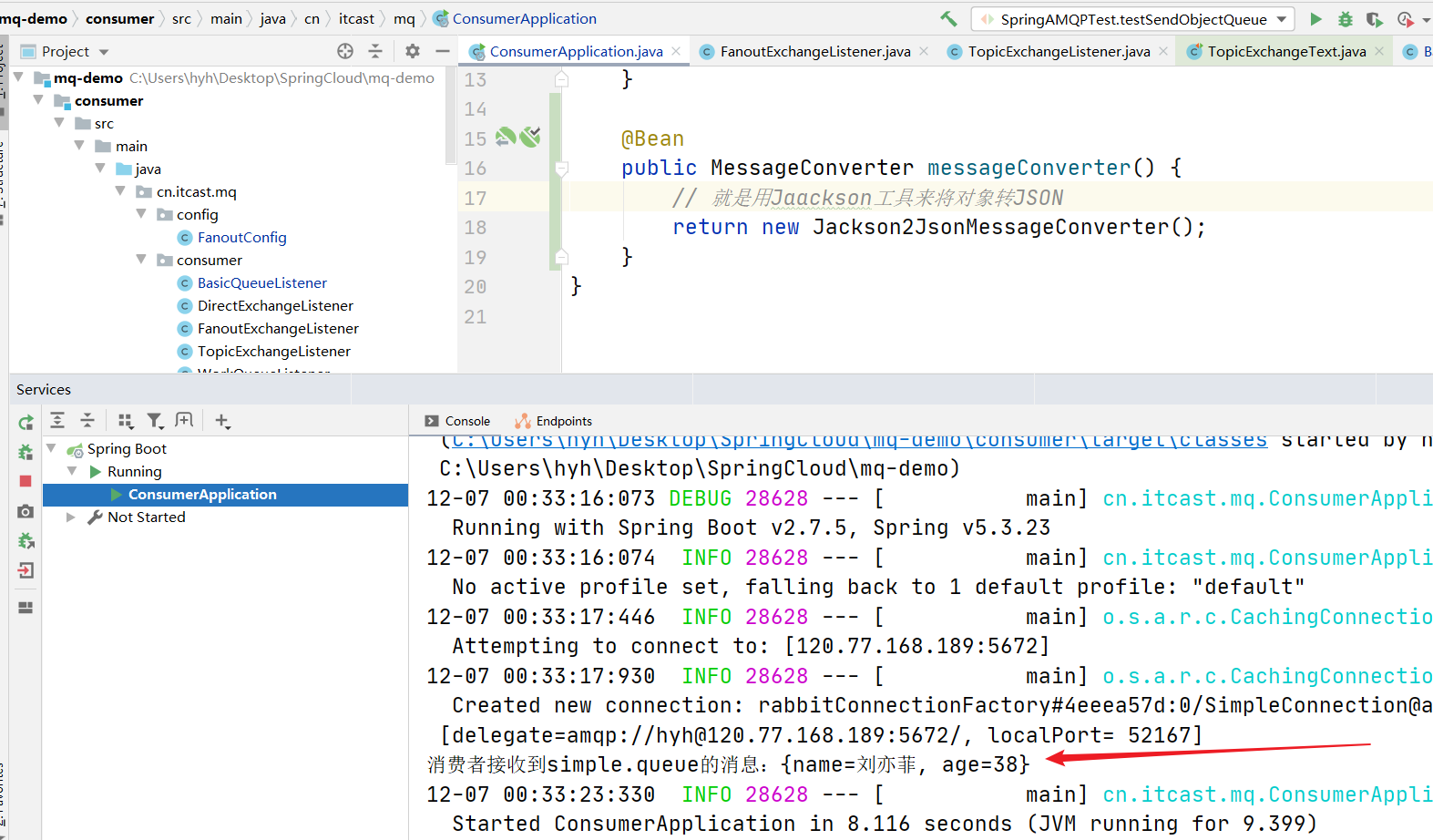
你砍你砍你砍，刘亦菲现原形了！！！



#### 5：consumer服务的监听器中消费消息（发送时什么对象，消费时就用什么对象接收）



#### 6：重启consumer，控制台查看消息消费日志：



## 总结：

SpringAMQP中消息的序列化和反序列化是怎么实现的？

1：利用MessageConverter实现，默认是JDK的序列化，推荐使用JSON方式的；

2：注意发送发与接收方必须使用相同的MessageConverter