

****

软 件 学 院

《中间件技术》大作业报告（一）

**题　　目 为ActiveMQ添加一个自定义的Message**

**姓　　名 陈澄**

**学　　号 32420212202930**

**班　　级 软工三班**

**实验时间 2024/04/25**

**2024 年 04 月 25 日**

# 实验题目

为ActiveMQ添加一个自定义的Message并支持其收发等主要功能

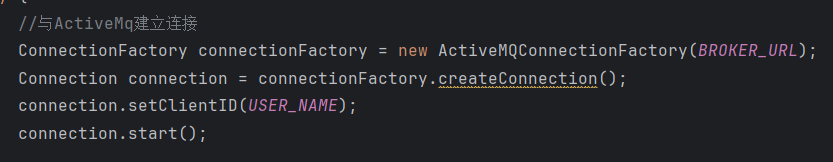
# 本次实验目的

阅读并ActiveMQ的代码并理解其消息收发流程与原理

# 实验步骤

1. 消息发送调用流程：

首先会创建一个ConnectionFactory类，传入代理的 URL（BROKER\_URL），该对象用于创建与代理的连接，并通过该类创建一个Connection对象。



其后，通过创建的Connection对象创建一个Session对象，此处会设置消息的确认模式。



创建一个Topic对象，作为对话的主题。



通过Topic创建一个MessageProducer，用于消息发送。



通过Session创建一个文本消息TextMessage。

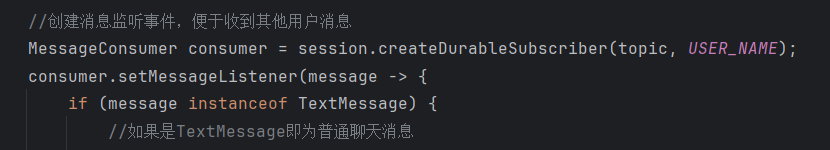


通过MessageProducer调用send()发送该消息。



1. 消息接收调用流程：

通过Session创建一个MessageConsumer对象，通过该对象创建一个消息监听器，根据收到的消息进行处理。



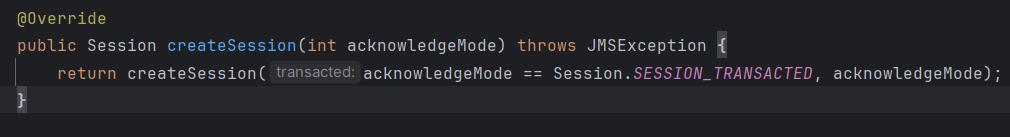
1. 消息发送代码：

（1）ConnectionFactory会将传入的String类型转化为URL类型并创建一个Connection对象返回。



createSession(int acknowledgeMode) 方法用于创建一个 JMS 会话对象。

acknowledgeMode 参数表示消息确认模式。

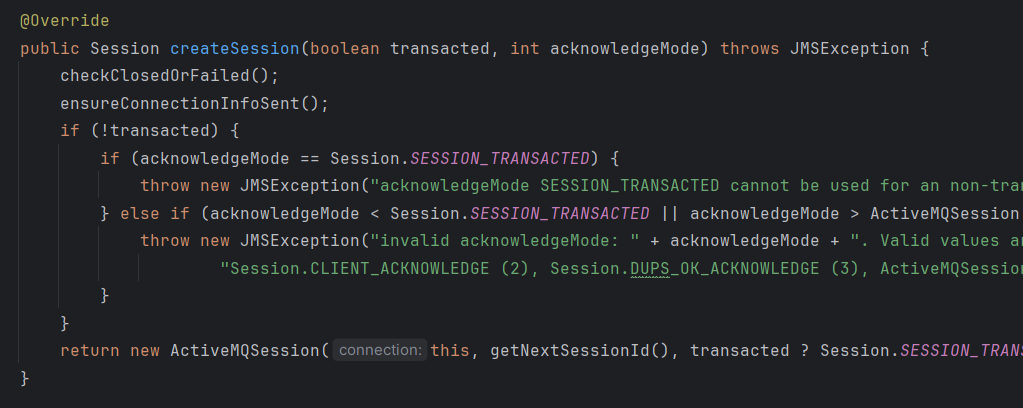


checkClosedOrFailed() 方法用于检查连接是否已关闭或失败，

ensureConnectionInfoSent() 方法用于确保连接信息已经被发送到代理，

isDispatchAsync() 方法用于判断是否异步分发消息，

isAlwaysSessionAsync() 方法用于判断是否总是使用异步会话。

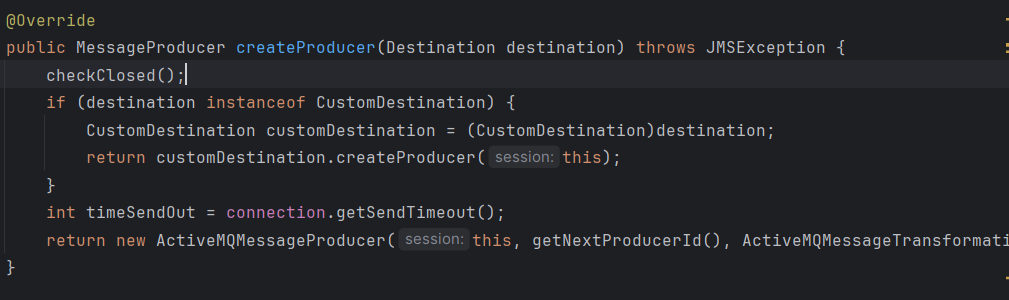


（2）createProducer(Destination destination)用于创建一个MessageProducer

同样先调用checkClosed()检查连接是否关闭

getSendTimeout()获得连接的失效时间

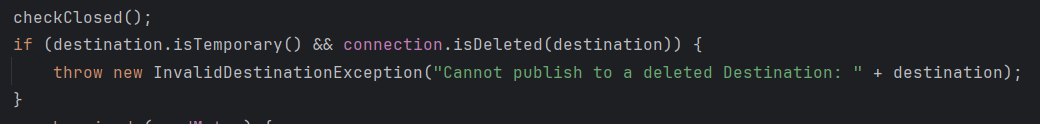
随后创建一个新的MessageProducer对象



（3）Send()方法用于消息的发送

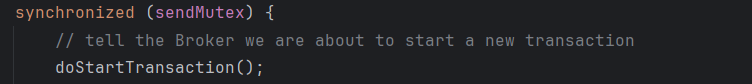
首先，会检查连接是否已关闭。

接下来，会检查目的地是否为临时目的地，并且该目的地是否已被删除。

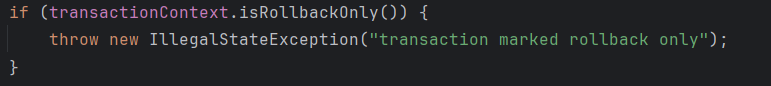


使用sendMutex对象进行同步，确保在发送消息时不会发生并发访问冲突。

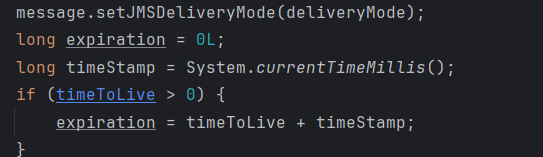
调用doStartTransaction()方法，告知Broker即将开始一个新的事务。



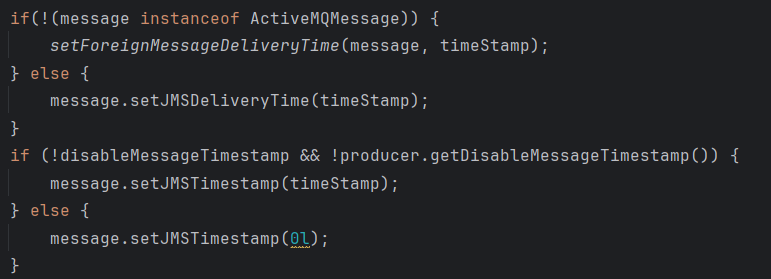
检查事务状态，事务不能被标记为回滚。



设置消息的JMS头字段，包括交付模式（deliveryMode）、过期时间（expiration）、时间戳（timeStamp）、优先级（priority）和是否重新传递（redelivered）。



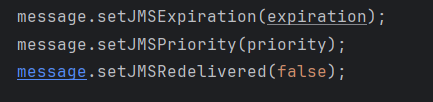
将原始消息转换为ActiveMQ消息格式。



根据禁用消息时间戳的设置，将消息的JMS时间戳（JMSTimestamp）设置为当前时间戳或0。

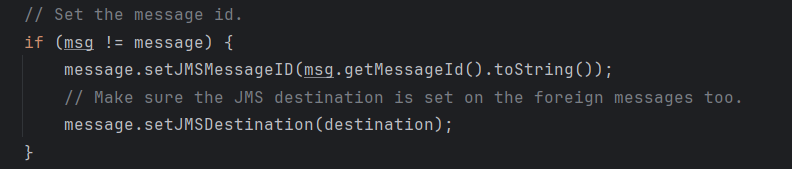
设置消息的过期时间和优先级。

将消息的JMS重新传递标志设置为false。

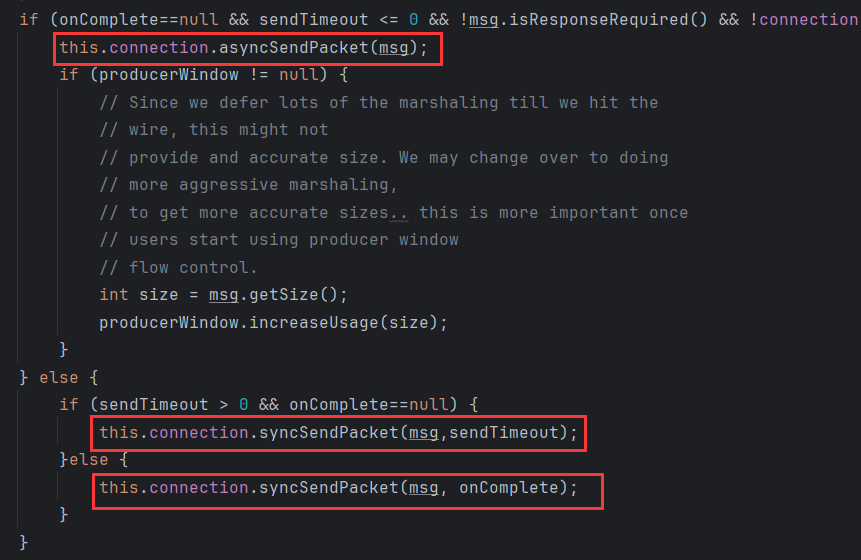


对消息进行转换，将其转换为自定义的消息格式（ActiveMQMessage），并将连接信息传递给消息。

设置消息的生产者ID（producerId），并进行日志记录。



根据发送超时时间（sendTimeout）和完成回调函数（onComplete），选择是异步还是同步地发送消息。



接着调用connection中的包发送方法进行发包。

1. 消息接收代码：

createConsumer()用于创建一个消息消费者

destination 参数表示消息消费的目标地址。

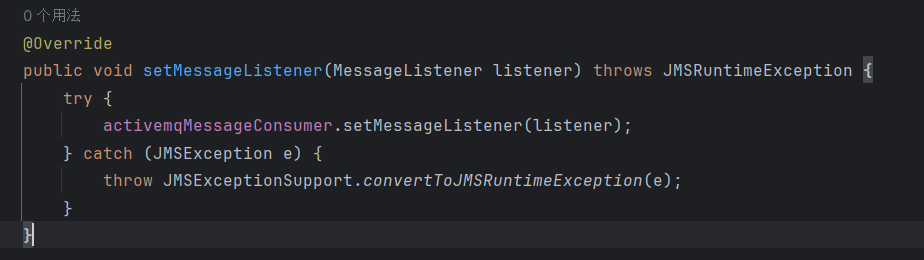
messageSelector 参数表示消息选择器，它是一个字符串，用于过滤要被消费的消息。

noLocal 参数表示是否禁止消费者接收自己发送的消息。

该方法会将jms中非custom的对象转化为custom的对象。



设置消息监听器则是直接调用jms中的消息监听器设置方法。

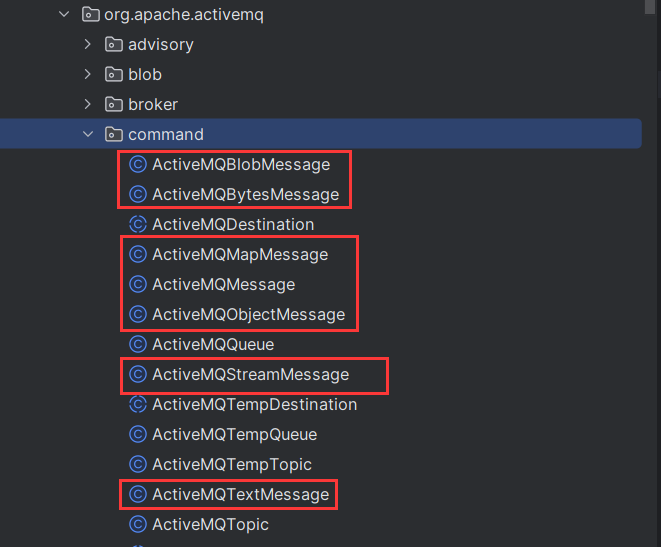


1. ActiveMQ中的Message

ActiveMQ中的ActiveMQMessage都继承自Message抽象类，并且实现org.apache.activemq.Message, ScheduledMessage两个接口。

在ActiveMQMessage实现了Message需要的主要方法。

其余各种Message都是继承ActiveMQMessage





例如：ActiveMQTextMessage则是继承ActiveMQMessage并实现jms中的TextMessage接口。

