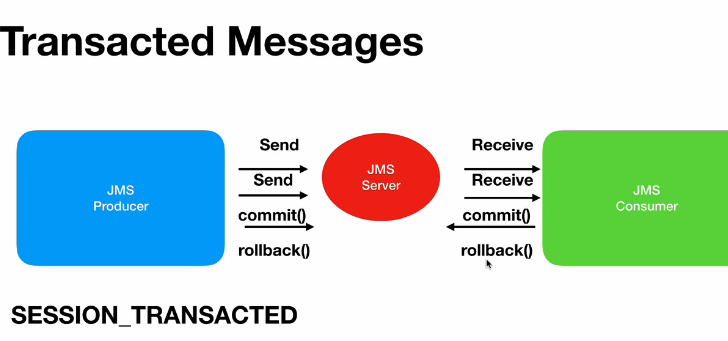
1. JMS Transactions

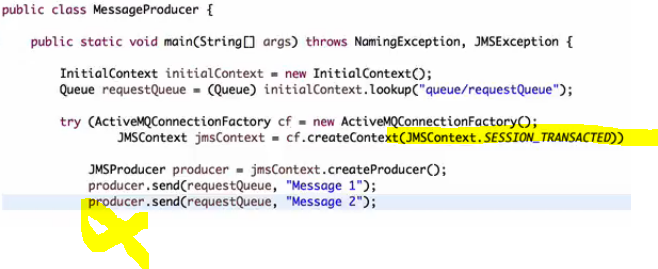


74.Comitting from producer side

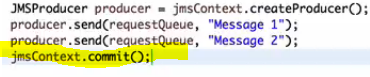
In this case,

Even though producer says “producer.send(queuename,data)” the actual message wont go the

Queue, it will be stored in the cache, once producer says commit, then only all the messages will go into the jms provider,



In the above case , none of the message will be delivered, because the producer didn’t committed the message



Use context.commit() to commit

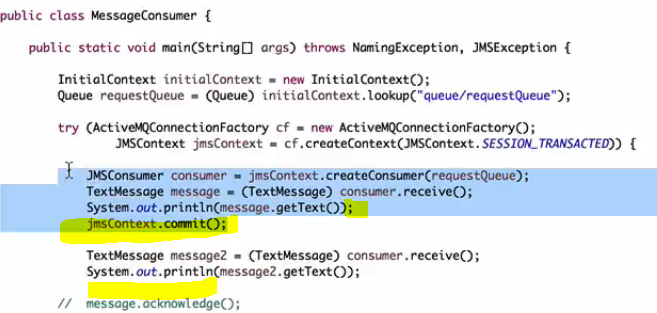
Once committed u can’t rollback,

75.comitting from consumer side

Similarly, even though consumer received the message, until the consumer says commit, the message will still present in the cache, jms think the message hasn’t been delivered, it will try to redeliver it again.

rollback

once rollbacked, all the messages present in the cache will be deleted and it wont go to the JMS server



In the above scenario, after receiving 1st message only, we are committing, means consumer is saying only 1 was received , bec he committed after receiving 1st message,

So 1st time, even though, both msgs were be received, but consumer sends acknowledgement only for 1 msg , 2nd once will still stays in the JMS Queue, when u run the same program for 2nd time it will still pull the message and then as we committed, then 2nd message also will be lost from JMS

80. Grouping the messages

Even after grouping, we will send those messages individually, but while sending we will send them with a selector, so that all messages will go into the same queue.

And while creating a consumer , we will create with that selector, so that only aatching records alone will be consumed by that queue.

**Even if multiple consumers are on listening queue, same group message will be received by same consumer**

