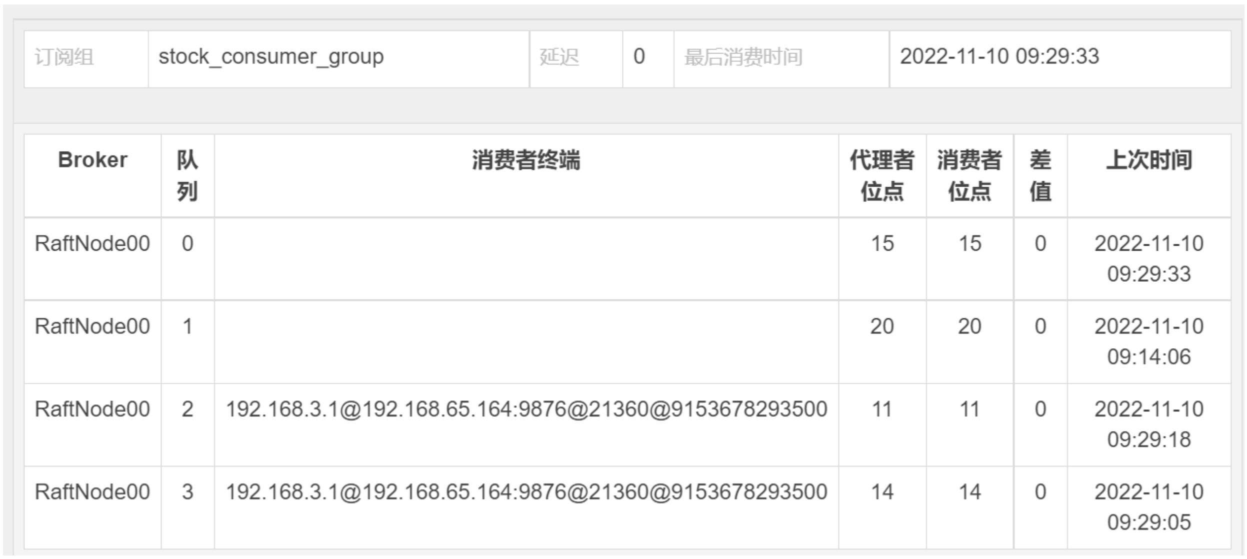
问题： 课上演示Rocketmq事务消息，本地事务执行完成提交后，消费端没有消费消息?

排查思路

1. 检查消费端topic配置是否正确
2. 打开Rocketmq控制台，查看topic的消费情况



订阅组	stock_consumer_group	延迟	0	最后消费时间	2022-11-10 09:29:33

Broker	队列	消费者终端	代理者位点	消费者位点	差值	上次时间
RaftNode00	0		15	15	0	2022-11-10 09:29:33
RaftNode00	1		20	20	0	2022-11-10 09:14:06
RaftNode00	2	192.168.3.1@192.168.65.164:9876@21360@9153678293500	11	11	0	2022-11-10 09:29:18
RaftNode00	3	192.168.3.1@192.168.65.164:9876@21360@9153678293500	14	14	0	2022-11-10 09:29:05

定位到问题所在：业务端消费者只订阅了broker部分队列，未订阅的队列的消息消费不到

原因： 启动了多个消费者

3. 排查是否启动了多个消费者，发现了问题所在。

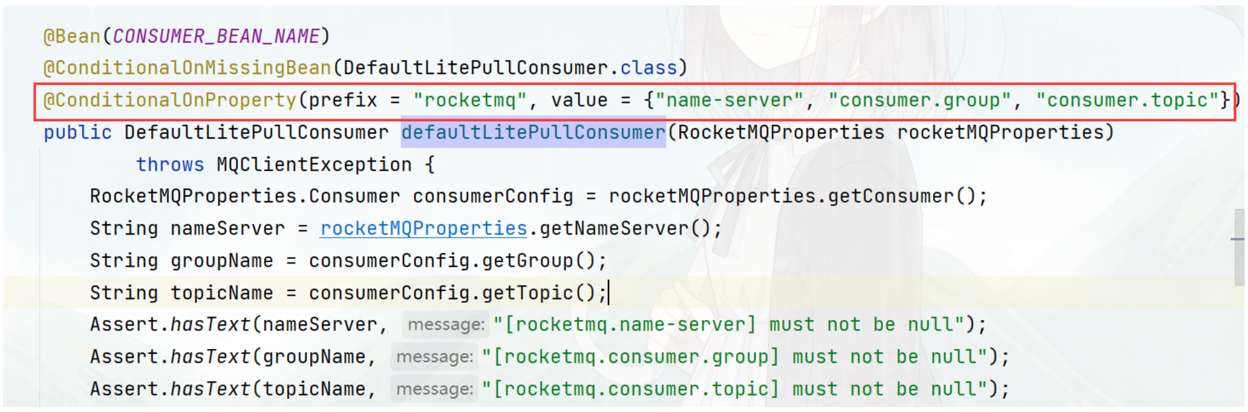
SpringBoot整合Rocketmq的坑，如果在yml中配置了如下配置，会默认创建一个消费者，导致业务类中配置的消费者无法消费部分broker队列的消息

```
rocketmq:
 name-server: 192.168.65.164:9876
 consumer:
 group: stock_consumer_group
 topic: reduce-stock
```

业务类中@RocketMQMessageListener指定消费组和topic，也会创建一个消费者

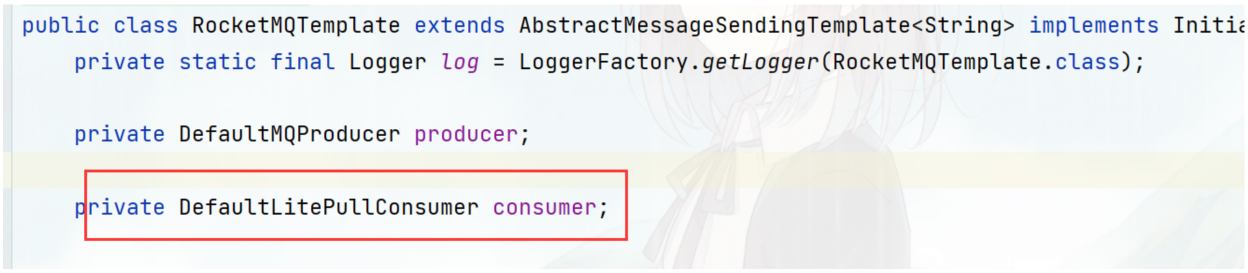
```
@Component
@RocketMQMessageListener(consumerGroup = "${rocketmq.consumer.group}",topic = "${rocketmq.consumer.topic}")
public class ReduceStockMsgConsumer implements RocketMListener<StockChangeEvent> {
```

源码：RocketMQAutoConfiguration#defaultLitePullConsumer



```
@Bean(CONSUMER_BEAN_NAME)
@ConditionalOnMissingBean(DefaultLitePullConsumer.class)
@ConditionalOnProperty(prefix = "rocketmq", value = {"name-server", "consumer.group", "consumer.topic"})
public DefaultLitePullConsumer defaultLitePullConsumer(RocketMQProperties rocketMQProperties)
        throws MQClientException {
    RocketMQProperties.Consumer consumerConfig = rocketMQProperties.getConsumer();
    String nameServer = rocketMQProperties.getNameServer();
    String groupName = consumerConfig.getGroup();
    String topicName = consumerConfig.getTopic();
    Assert.hasText(nameServer, message: "[rocketmq.name-server] must not be null");
    Assert.hasText(groupName, message: "[rocketmq.consumer.group] must not be null");
    Assert.hasText(topicName, message: "[rocketmq.consumer.topic] must not be null");
```

DefaultLitePullConsumer会用于RocketMQTemplate接收消息



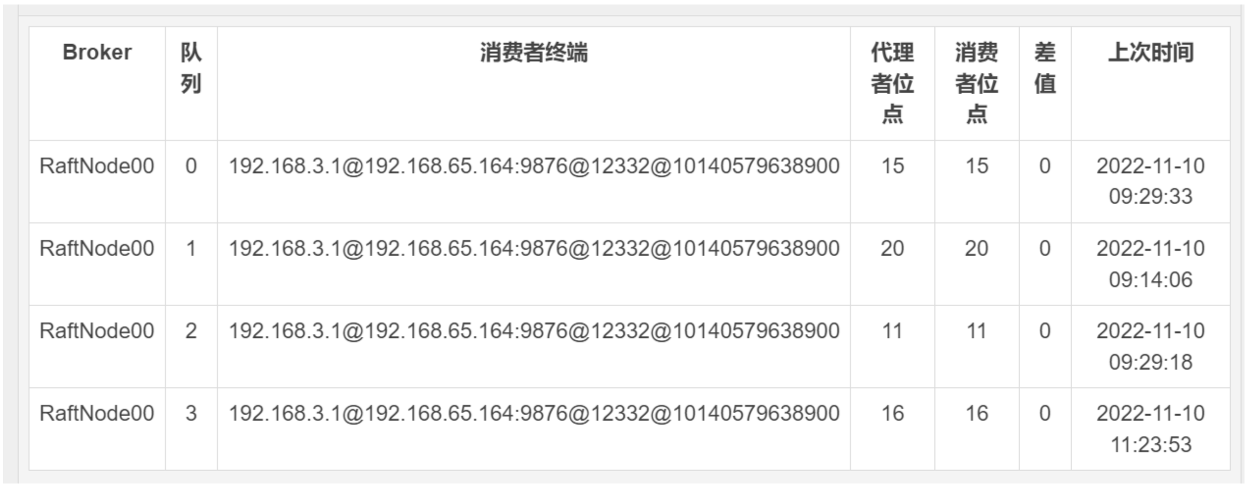
```
public class RocketMQTemplate extends AbstractMessageSendingTemplate<String> implements Initia
    private static final Logger log = LoggerFactory.getLogger(RocketMQTemplate.class);

    private DefaultMQProducer producer;

    private DefaultLitePullConsumer consumer;
```

4. 修改yml配置并修改业务代码@RocketMQMessageListener配置

```
rocketmq:
 name-server: 192.168.65.164:9876
 stock_consumer:
 group: stock_consumer_group
 topic: reduce-stock
```

```
@Component
//@RocketMQMessageListener(consumerGroup = "stock_consumer_group",topic =
"reduce-stock")
@RocketMQMessageListener(consumerGroup = "${rocketmq.stock_consumer.grou
p}",topic = "${rocketmq.stock_consumer.topic}")
public class ReduceStockMsgConsumer implements RocketMQListener<StockChan
geEvent> {
```

5. 重启服务后查看topic情况



Broker	队列	消费者终端	代理者位点	消费者位点	差值	上次时间
RaftNode00	0	192.168.3.1@192.168.65.164:9876@12332@10140579638900	15	15	0	2022-11-10 09:29:33
RaftNode00	1	192.168.3.1@192.168.65.164:9876@12332@10140579638900	20	20	0	2022-11-10 09:14:06
RaftNode00	2	192.168.3.1@192.168.65.164:9876@12332@10140579638900	11	11	0	2022-11-10 09:29:18
RaftNode00	3	192.168.3.1@192.168.65.164:9876@12332@10140579638900	16	16	0	2022-11-10 11:23:53

可以看到消费端已经订阅所有的broker了

rocketmq事务消息演示.mp4
3.67MB