

图文 94 如何基于本地运行的RocketMQ进行消息的生产与消费?

399 人次阅读 2020-02-07 09:22:54

[详情](#) [评论](#)

狸猫技术

如何基于本地运行的RocketMQ进行消息的生产与消费?**相关频道**

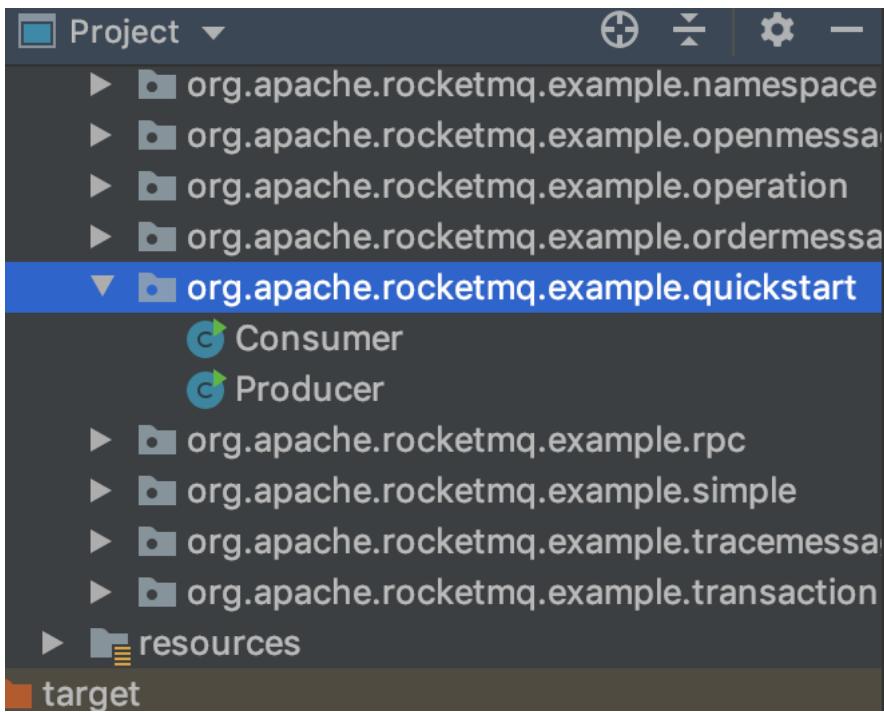
继《从零开始带你成为JVM实战高手》后，又一重磅力作

(点击下方蓝字试听)

[《从零开始带你成为MySQL实战优化高手》](#)**1、使用RocketMQ自带的例子程序测试消息的发送和消费**

接着我们可以使用RocketMQ自带的例子程序来测试消息的发送和消费，就在example模块下面就有，我们看下面的图。

在图中我们可以在quickstart包下找到Producer和Consumer两个例子类，而且大家可以看到，在这里包括事务消息、顺序消息，等等，很多高阶功能的例子在这里都有。



2、创建一个测试用的Topic出来

首先我们需要启动rocketmq-console工程，之前在29讲里教过大家启动这个rocketmq控制台的步骤，大家把这个工程启动即可，在启动之后，我们就可以看到集群里有一台broker机器，如下图。

Broker	NO.	Address	Version	Produce Massage TPS	Consumer Massage TPS	Yesterday Produce Count	Yesterday Consume Count	Today Produce Count	Today Consume Count	Operation
broker-a	0(master)	192.168.3.9:10911	V4_6_0	0.00	0.00	0	0	1	0	<button>STATUS</button> <button>CONFIG</button>

接着我们就进入Topic菜单，新建一个名称为TopicTest的Topic即可，新建完之后在Topic列表就可以看到下面的内容了。

Topic	Operation
RMQ_SYS_TRANS_HALF_TOPIC	<button>STATUS</button> <button>ROUTER</button> <button>CONSUMER MANAGE</button> <button>TOPIC CONFIG</button> <button>SEND MESSAGE</button> <button>RESET CONSUMER OFFSET</button> <button>DELETE</button>
TopicTest	<button>STATUS</button> <button>ROUTER</button> <button>CONSUMER MANAGE</button> <button>TOPIC CONFIG</button> <button>SEND MESSAGE</button> <button>RESET CONSUMER OFFSET</button> <button>DELETE</button>

3、修改和运行RocketMQ自带的Producer示例程序

接着我们来修改一下RocketMQ自带的Producer示例程序，如下所示：

```
public class Producer {  
    public static void main(String[] args)  
        throws MQClientException, InterruptedException {  
        /*  
         * Instantiate with a producer group name.  
         */  
        DefaultMQProducer producer = new DefaultMQProducer("please_rename_unique_group_name");  
        /*  
         * Specify name server addresses.  
         * <p/>  
         * Alternatively, you may specify name server addresses via exporting environmental variable: NAMESRV_ADDR  
         * <pre>  
         * {@code  
         *     producer.setNamesrvAddr("name-server1-ip:9876;name-server2-ip:9876");  
         * }  
         * </pre>  
         */  
        // 最核心的是修改这里，要设置一下NameServer的地址才行  
        // 要确保Producer可以找到NameServer去获取当前的Broker地址  
        producer.setNamesrvAddr("127.0.0.1:9876");  
        /*  
         * Launch the instance.  
         */  
        producer.start();  
    }  
}
```

```

// 这里原来是发送1000条消息，我们修改为发送1条消息就可以了
for (int i = 0; i < 1; i++) {
    try {
        /*
         * Create a message instance, specifying topic, tag and message body.
         */
        Message msg = new Message("TopicTest" /* Topic */,
                                  "TagA" /* Tag */,
                                  ("Hello RocketMQ " + i).getBytes(RemotingHelper.DEFAULT_CHARSET) /* Message body */
        );
        /*
         * Call send message to deliver message to one of brokers.
         */
        SendResult sendResult = producer.send(msg);
        System.out.printf("%s%n", sendResult);
    } catch (Exception e) {
        e.printStackTrace();
        Thread.sleep(1000);
    }
}
/*
 * Shut down once the producer instance is not longer in use.
 */
producer.shutdown();
}
}

```

接着我们执行运行上面的程序就可以了，他会发送1条消息到Broker里去，我们观察一下控制台的日志打印，可以看到下面的内容，就说明我们已经成功的把消息发送到Broker里去了。

```

SendResult [sendStatus=SEND_OK, msgId=240E03A24CD1B7A0B066027402ACC71F000018B4AAC217E3F1580000,
offsetMsgId=C0A8030900002A9F0000000000000000, messageQueue=MessageQueue [topic=TopicTest,
brokerName=broker-a, queueId=2], queueOffset=0]

```

4、修改和运行RocketMQ自带的Consume示例程序

接着修改RocketMQ自带的Consumer示例程序：

```

public class Consumer {

    public static void main(String[] args)
        throws InterruptedException, MQClientException {
        /*
         * Instantiate with specified consumer group name.
         */
        DefaultMQPushConsumer consumer = new DefaultMQPushConsumer("please_rename_unique_group_name_4");
        /*
         * Specify name server addresses.
         * <p/>
         * Alternatively, you may specify name server addresses via exporting environmental variable: NAMESRV_ADDR
         * <pre>
         * {@code
         * consumer.setNamesrvAddr("name-server1-ip:9876;name-server2-ip:9876");
         * }
         * </pre>
         */
        // 重点就是这里，要设置一下nameserver的地址
        consumer.setNamesrvAddr("127.0.0.1:9876");
    }
}

```

```

/*
 * Specify where to start in case the specified consumer group is a brand new one.
 */
consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);

/*
 * Subscribe one or more topics to consume.
 */
consumer.subscribe("TopicTest", "*");

/*
 * Register callback to execute on arrival of messages fetched from brokers.
 */
consumer.registerMessageListener(new MessageListenerConcurrently() {
    @Override
    public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt> msgs,
        ConsumeConcurrentlyContext context) {
        System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
        return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
    }
});

/*
 * Launch the consumer instance.
 */
consumer.start();
System.out.printf("Consumer Started.%n");
}
}
}

```

接着运行上述程序，可以看到消费到了1条消息，如下所示：

```

ConsumeMessageThread_1 Receive New Messages: [MessageExt [queueId=2, storeSize=225, queueOffset=0, sysFlag=0,
bornTimestamp=1580887214424, bornHost=/192.168.3.9:56600, storeTimestamp=1580887214434,
storeHost=/192.168.3.9:10911, msgId=C0A8030900002A9F0000000000000000, commitLogOffset=0, bodyCRC=613185359,
reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message{topic='TopicTest', flag=0, properties=
{MIN_OFFSET=0, MAX_OFFSET=1, CONSUME_START_TIME=1580887519080,
UNIQ_KEY=240E03A24CD1B7A0B066027402ACC71F000018B4AAC217E3F1580000, CLUSTER=DefaultCluster, WAIT=true,
TAGS=TagA}, body=[72, 101, 108, 108, 111, 32, 82, 111, 99, 107, 101, 116, 77, 81, 32, 48], transactionId='null'}]]

```

到此为止，我们的RocketMQ的源码环境彻底搭建完毕了，而且可以在本地启动以及收发消息，其实到这里为止，我们就可以去调试和分析RocketMQ的源码了。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐：

[《从零开始带你成为JVM实战高手》](#)
[《21天互联网Java进阶面试训练营》（分布式篇）](#)
[《互联网Java工程师面试突击》（第1季）](#)
[《互联网Java工程师面试突击》（第3季）](#)

重要说明：

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《付费用户如何加群》（[购买后可见](#)）

Copyright © 2015-2020 深圳小鹅网络技术有限公司 All Rights Reserved. [粤ICP备15020529号](#)

 小鹅通提供技术支持